



Leitfaden

AWS Schema Conversion Tool



Version 1.0.672

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Schema Conversion Tool: Leitfaden

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Die Marken und Handelsmarken von Amazon dürfen nicht in einer Weise in Verbindung mit nicht von Amazon stammenden Produkten oder Services verwendet werden, die geeignet ist, Kunden irrezuführen oder Amazon in irgendeiner Weise herabzusetzen oder zu diskreditieren. Alle anderen Handelsmarken, die nicht Eigentum von Amazon sind, gehören den jeweiligen Besitzern, die möglicherweise zu Amazon gehören oder nicht, mit Amazon verbunden sind oder von Amazon gesponsert werden.

Table of Contents

Was ist AWS SCT?	1
Übersicht über die Schemakonvertierung	5
Feedback geben	7
Installation, Überprüfung und Aktualisierung	8
Installieren AWS SCT	8
Der AWS SCT Dateidownload wird überprüft	10
Überprüfung der Prüfsumme der Datei AWS SCT	10
Überprüfung der AWS SCT RPM-Dateien auf Fedora	11
Überprüfung der AWS SCT DEB-Dateien auf Ubuntu	12
Überprüfen der AWS SCT MSI-Datei unter Microsoft Windows	12
Die erforderlichen Datenbanktreiber werden heruntergeladen	13
Installation von JDBC-Treibern unter Linux	16
Speichern von Treiberpfaden in den globalen Einstellungen	17
Aktualisierung AWS SCT	18
AWS SCT CLI	19
Verwendung der AWS SCT Benutzeroberfläche	20
Das Projektfenster	20
Starten von AWS SCT	22
Erstellen eines Projekts	22
Verwenden eines Assistenten für neue Projekte	23
Projekt speichern und öffnen	27
Einen Server hinzufügen	28
Verwenden eines Offline-Modus	29
Verwenden von Baumfiltern	30
.....	31
Import einer Dateiliste für den Baumfilter	33
Schemas ausblenden	34
Verwaltung des Bewertungsberichts zur Datenbankmigration	36
Konvertieren Ihres Schemas	40
Den konvertierten Code anwenden	43
AWSProfile speichern	44
AWSAnmeldeinformationen speichern	45
Festlegen des Standardprofils für ein Projekt	47
Berechtigungen für die Verwendung des AWS Dienstprofils	47

Verwenden von AWS Secrets Manager	48
Speichern von Datenbankpasswörtern	49
Verwenden der Ansicht „Alle verbinden“ für Projekte mit partitionierten Tabellen	49
Tastenkombinationen	50
Erste Schritte	52
Quellen für AWS SCT	54
Verschlüsselung von Amazon RDS-Verbindungen	55
Apache Cassandra als Quelle verwenden	58
Verbindung zu Apache Cassandra als Quelle herstellen	58
Apache Hadoop als Quelle verwenden	60
Voraussetzungen für die Verwendung von Apache Hadoop als Quelle	61
Berechtigungen für Hive als Quelle	62
Berechtigungen für HDFS als Quelle	62
Berechtigungen für HDFS als Ziel	63
Verbindung zu Apache Hadoop als Quelle herstellen	63
Verbindung zu Hive und HDFS herstellen	65
Verbindung zu Amazon EMR als Ziel herstellen	68
Apache Oozie als Quelle verwenden	71
Voraussetzungen	72
Verbindung zu Apache Oozie als Quelle herstellen	72
Berechtigungen für AWS Lambda	74
AWS Step FunctionsAls Ziel eine Verbindung herstellen	76
Azure SQL Database als Quelle verwenden	77
Berechtigungen für Azure SQL Database	78
Verbindung zur Azure SQL-Datenbank als Quelle herstellen	78
IBM Db2 for z/OS als Quelle verwenden	80
Voraussetzungen für Db2 for z/OS	80
Rechte für Db2 for z/OS	81
Verbindung zu Db2 für z/OS als Quelle herstellen	82
Privilegien für MySQL als Ziel	84
Privilegien für PostgreSQL als Ziel	86
Einstellungen für die Konvertierung von Db2 für z/OS in PostgreSQL	87
Verwenden von IBM Db2 LUW als Quelle	88
Rechte für Db2 LUW	89
Verbindung zu Db2 LUW als Quelle herstellen	91
Db2 LUW nach PostgreSQL	94

Db2 LUW nach MySQL	96
Verwenden von MySQL als Quelle	98
Rechte für MySQL	98
Verbindung zu MySQL als Quelle	98
Privilegien für PostgreSQL als Ziel	101
Oracle Database als Quelle verwenden	102
Privilegien für Oracle	103
Verbindung zu Oracle als Quelle herstellen	103
Oracle zu PostgreSQL	109
Oracle in MySQL	115
Von Oracle zu Amazon RDS für Oracle	125
Verwenden von PostgreSQL als Quelle	132
Privilegien für PostgreSQL	132
Herstellen einer Verbindung zu PostgreSQL als Quelle	133
Privilegien für MySQL als Ziel	135
SAP ASE (Sybase ASE) als Quelle verwenden	137
Privilegien für SAP ASE	137
Verbindung zu SAP ASE als Quelle herstellen	138
Privilegien für MySQL als Ziel	140
Einstellungen für die Konvertierung von SAP ASE zu MySQL	142
Privilegien für PostgreSQL als Ziel	143
Einstellungen für die Konvertierung von SAP ASE zu PostgreSQL	143
Verwenden von SQL Server als Quelle	145
Rechte für die Microsoft SQL -Prozedur.	145
Die Windows-Authentifizierung mit Microsoft SQL Server verwenden.	147
Verbindung mit SQL Server als -Quelle herstellen.	149
SQL Server in MySQL	152
SQL Server in PostgreSQL	157
SQL Server in Amazon RDS SQL Server	193
Data Warehouse-Quellen für AWS SCT	195
Amazon Redshift als Quelle verwenden	195
Azure Synapse Analytics als Quelle verwenden	201
Verwenden von BigQuery als Quelle	207
Greenplum Database als Quelle verwenden	213
Netezza als Quelle verwenden	220
Oracle Data Warehouse als Quelle verwenden	230

Snowflake als Quelle verwenden	238
SQL Server Data Warehouse als Quelle verwenden	247
Teradata als Quelle verwenden	254
Vertica als Quelle verwenden	270
Mapping-Regeln erstellen	278
Neue Regel	279
Regeln verwalten	279
Virtuelle Ziele	281
Einschränkungen	282
Erstellen von Konvertierungsberichten	283
Berichte über die Migrationsbewertung	283
Erstellen eines Berichts über die Bewertung der Datenbankmigration	284
Den Bewertungsbericht anzeigen	285
Speichern des Bewertungsberichts	289
Konfiguration des Bewertungsberichts	292
Einen Bewertungsbericht für mehrere Server erstellen	295
Datenbankschemas konvertieren	305
Migrationsregeln erstellen	307
Migrationsregeln erstellen	308
Migrationsregeln exportieren	310
Ihr Schema konvertieren	310
Konvertieren des Schemas	311
Konvertiertes Schema bearbeiten	313
Löschen eines konvertierten Schemas	315
Umgang mit manuellen Konvertierungen	315
Ändern Ihres Quellschemas	315
Ihr Zielschema ändern	315
Aktualisierung und Aktualisierung Ihres konvertierten Schemas	316
Speichern und Anwenden Ihres Schemas	317
Speichern Ihres konvertierten Schemas	317
Ihr konvertiertes Schema anwenden	318
Das Schema des Erweiterungspakets	319
Schemas vergleichen	319
Verwandte transformierte Objekte	321
Konvertierung von Data Warehouse-Schemas nach Amazon Redshift	322
Berechtigungen für Amazon Redshift	324

Auswahl von Optimierungsstrategien und -regeln	325
Statistiken sammeln oder hochladen	327
Migrationsregeln erstellen	328
Migrationsregeln erstellen	329
Migrationsregeln exportieren	331
Ihr Schema konvertieren	331
Konvertieren des Schemas	332
Konvertiertes Schema bearbeiten	334
Löschen eines konvertierten Schemas	335
Schlüssel verwalten und anpassen	335
Verwandte Themen	336
Erstellung und Verwendung des Bewertungsberichts	336
Erstellen eines Bewertungsberichts zur Datenbankmigration	336
Übersicht	337
Aktionspunkte	339
Speichern des Bewertungsberichts	339
Umgang mit manuellen Konvertierungen	340
Ändern Ihres Quellschemas	341
Ihr Zielschema ändern	341
Aktualisierung und Aktualisierung Ihres konvertierten Schemas	342
Speichern und Anwenden Ihres konvertierten Schemas	342
Speichern Ihres konvertierten Schemas in einer Datei	343
Wenden Sie Ihr konvertiertes Schema an	344
Das Schema des Erweiterungspakets	344
Python-Bibliotheken	345
Optimierung von Amazon Redshift	345
Optimierung Ihrer Amazon Redshift-Datenbank	345
ETL-Prozesse konvertieren	348
Konvertierung von ETL-Prozessen inAWS Glue	349
Voraussetzungen	350
AWS Glue-Datenkatalog	351
Einschränkungen	351
Schritt 1: Erstellen eines neuen -Projekts	353
Schritt 2: Erstellen Sie eineAWS GlueAufgabe	354
Konvertierung von ETL-Prozessen mithilfe der Python-API fürAWS Glue	356
Schritt 1: Erstellen einer Datenbank	356

Schritt 2: Verbindung erstellen	357
Schritt 3: Erstellen Sie eineAWS GlueCrawler	358
Konvertierung von Informatica ETL-Skripten	361
SSIS konvertieren inAWS Glue	366
Unterstützte SSIS-Komponenten	370
SSIS konvertieren inAWS Glue Studio	372
Voraussetzungen	372
Hinzufügen von SSIS-Paketen zu IhremAWS SCTProjekt	374
SSIS-Pakete konvertieren	375
ErstellenAWS Glue StudioJobs	376
Erstellen eines SSIS-Konvertierungsbewertungsberichts	378
Unterstützte SSIS-Komponenten	379
Konvertierung von Teradata BTEQ zu Amazon Redshift RSQL	380
Hinzufügen von BTEQ-Skripten zu IhremAWS SCTProjekt	381
Konfiguration von Substitutionsvariablen in BTEQ-Skripten	382
Konvertierung von BTEQ-Skripten	383
Verwaltung von BTEQ-Skripten	384
Erstellen eines Bewertungsberichts zur BTEQ-Skriptkonvertierung	384
Bearbeiten und Speichern Ihrer konvertierten BTEQ-Skripte	385
Konvertierung von Shell-Skripten nach Amazon Redshift RSQL	386
Hinzufügen von Shell-Skripten zu IhremAWS SCTProjekt	386
Konfiguration von Substitutionsvariablen in Shell-Skripten	387
Shell-Skripte konvertieren	388
Shell-Skripte verwalten	389
Einen Bericht zur Bewertung der Shell-Skript-Konvertierung erstellen	390
Bearbeiten und Speichern Ihrer konvertierten Shell-Skripte	391
Teradata konvertierenFastExportzu Amazon Redshift RSQL	391
HinzufügenFastExportJob-Skripte zu IhremAWS SCTProjekt	392
Konfiguration von Substitutionsvariablen inFastExportJob-Skripte	393
KonvertierenFastExportJob-Skripte	394
ManagenFastExportJob-Skripte	395
Erstellen einesFastExportBericht zur Bewertung der Konvertierung von Jobskripten	396
Bearbeiten und Speichern Ihrer konvertiertenFastExportJob-Skripte	397
Teradata konvertierenFastLoadzu Amazon Redshift RSQL	397
HinzufügenFastLoadJob-Skripte zu IhremAWS SCTProjekt	398
Konfiguration von Substitutionsvariablen inFastLoadJob-Skripte	399

KonvertierenFastLoadJob-Skripte	400
ManagenFastLoadJob-Skripte	401
Erstellen einesFastLoadBericht zur Bewertung der Konvertierung von Jobskripten	402
Bearbeiten und Speichern Ihrer konvertiertenFastLoadJob-Skripte	403
Teradata konvertierenMultiLoadzu Amazon Redshift RSQL	403
HinzufügenMultiLoadJob-Skripte zu IhremAWS SCTProjekt	404
Konfiguration von Substitutionsvariablen inMultiLoadJob-Skripte	405
KonvertierenMultiLoadJob-Skripte	406
ManagenMultiLoadJob-Skripte	407
Erstellen einesMultiLoadBericht zur Bewertung der Konvertierung von Jobskripten	408
Bearbeiten und Speichern Ihrer konvertiertenMultiLoadJob-Skripte	409
Migration von Big-Data-Frameworks	410
Migration von Apache Hadoop zu Amazon EMR	410
Übersicht	411
Schritt 1: Stellen Sie eine Verbindung zu Ihren Hadoop-Clustern her	412
Schritt 2: Richten Sie die Zuordnungsregeln ein	412
Schritt 3: Erstellen Sie einen Bewertungsbericht	414
Schritt 4: Migrieren Sie Ihren Apache Hadoop-Cluster zu Amazon EMR	416
Ihr CLI-Skript ausführen	417
Verwaltung Ihres Migrationsprojekts	417
Konvertierung von Apache Oozie zu AWS Step Functions	419
Übersicht	419
Schritt 1: Stellen Sie eine Verbindung zu Ihren Quell- und Zieldiensten her	421
Schritt 2: Richten Sie die Zuordnungsregeln ein	422
Schritt 3: Parameter konfigurieren	422
Schritt 4: Erstellen Sie einen Bewertungsbericht	424
Schritt 5: Konvertieren Sie Ihre Apache Oozie-Workflows in AWS Step Functions	426
Ihr CLI-Skript ausführen	429
Unterstützte Knoten	429
Verwenden von AWS SCT mit AWS DMS	431
Verwendung einerAWS SCTReplikationsagenten mitAWS DMS	431
Verwendung einerAWS SCTDatenextraktionsagent mitAWS DMS	431
Erhöhung der Protokollierungsstufen bei VerwendungAWS SCTmitAWS DMS	431
Migration von einem Data Warehouse zu Amazon Redshift	433
Voraussetzungen	436
Amazon S3 S3-Einstellungen	436

Übernahme von IAM-Rollen	437
Sicherheitseinstellungen	439
Konfigurationseinstellungen	440
Installieren von -Agenten	440
Agenten konfigurieren	442
Installation und Konfiguration von speziellen Kopieragenten	444
Agenten starten	446
Registrierung von Agenten	447
Informationen für einen AWS SCT Agenten ausblenden und wiederherstellen	448
Regeln für die Datenmigration erstellen	450
Ändern der Extraktor- und Kopiereinstellungen für die Datenmigration	451
Daten sortieren	454
Eine AWS SCT Aufgabe erstellen, ausführen und überwachen	456
Exportieren und Importieren einer Datenextraktionsaufgabe	460
Datenextraktion mit einem AWS Snowball Edge-Gerät	461
tep-by-step S-Verfahren für die Datenmigration mit AWS SCT und Edge AWS Snowball	462
Ausgabe der Datenextraktionsaufgabe	465
Verwenden von virtueller Partitionierung	467
Einschränkungen bei der Erstellung einer virtuellen Partitionierung	467
RANGE-Partitionstyp	468
LIST-Partitionstyp	469
DATE AUTO SPLIT Partitionstyp	470
Verwenden der nativen Partitionierung	471
Arbeiten mit LOBs	473
Bewährte Methoden und Problemlösung	474
Anwendungs-SQL konvertieren	476
Überblick über die Konvertierung von Anwendungs-SQL	476
Konvertieren von SQL-Code in Ihren Anwendungen	477
Projekte zur Konvertierung generischer Anwendungen erstellen	477
Verwaltung von Projekten zur Konvertierung von Anwendungen	482
Ihren SQL-Code analysieren und konvertieren	483
Erstellung und Verwendung des Bewertungsberichts	484
Bearbeiten und Speichern Ihres konvertierten SQL-Codes	485
Konvertieren von SQL-Code in C#-Anwendungen	486
Projekte zur Konvertierung von C#-Anwendungen erstellen	486
Konvertieren Sie den SQL-Code Ihrer C#-Anwendung	487

Speichern Sie Ihren konvertierten Anwendungscode	489
Verwaltung von Projekten zur Konvertierung von C#-Anwendungen	489
Erstellen eines Bewertungsberichts zur C#-Anwendungskonvertierung	490
Konvertieren von SQL-Code in C++-Anwendungen	491
Projekte zur Konvertierung von C++-Anwendungen erstellen	492
Konvertieren Sie den SQL-Code Ihrer C++-Anwendung	493
Speichern Sie Ihren konvertierten Anwendungscode	495
Verwaltung von Projekten zur Konvertierung von C++-Anwendungen	496
Erstellen eines Bewertungsberichts zur Konvertierung von C++-Anwendungen	498
Umwandeln von SQL-Code in Java-Anwendungen	499
Projekte zur Konvertierung von Java-Anwendungen erstellen	499
Konvertieren Sie den SQL-Code Ihrer Java-Anwendung	501
Speichern Sie Ihren konvertierten Anwendungscode	503
Verwaltung von Projekten zur Konvertierung von Java-Anwendungen	503
Erstellen eines Bewertungsberichts zur Konvertierung von Java-Anwendungen	504
Konvertieren von SQL-Code in Pro*C-Anwendungen	506
Projekte zur Konvertierung von Pro*C-Anwendungen erstellen	506
Konvertieren Sie den SQL-Code Ihrer Pro*C-Anwendung	508
Bearbeiten und Speichern Ihres konvertierten Anwendungscode	509
Verwaltung von Projekten zur Konvertierung von Pro*C-Anwendungen	510
Erstellen eines Bewertungsberichts zur Konvertierung von Pro*C-Anwendungen	511
Verwenden von Erweiterungspaketen	514
Berechtigungen für die Verwendung des Erweiterungspakets	516
Verwenden des Erweiterungspaket-Schemas	517
Benutzerdefinierte Bibliotheken für Erweiterungspakete	518
Anwenden des Erweiterungspakets	518
Verwenden der Lambda-Funktionen aus dem AWS SCT Erweiterungspaket	521
Verwendung von AWS Lambda Funktionen zur Emulierung von Datenbankfunktionen	521
Anwenden des Erweiterungspakets zur Unterstützung von Lambda-Funktionen	521
Konfiguration der Funktionen des Erweiterungspakets	523
Bewährte Methoden	525
Konfigurieren zusätzlichen	525
Standard-Projektordner	525
Erhöhung der Datenmigration	526
Erhöhung der Protokollierungsinformationen	526
Fehlerbehebung	530

Objekte können nicht aus einer Oracle-Quelldatenbank geladen werden.	530
WarWarWarWarWar	530
CLI Reference	532
Voraussetzungen	532
interaktiver Modus	532
Beispiele	534
CLI-Szenarien abrufen	534
Beispiele	539
CLI-Szenarien bearbeiten	539
Skriptmodus	541
Beispiele	542
Referenzmaterial	542
Versionshinweise	543
Versionshinweise — 675	543
Versionshinweise — 674	545
Versionshinweise — 673	553
Versionshinweise — 672	557
Versionshinweise — 671	566
Versionshinweise — 670	575
Versionshinweise — 669	580
Versionshinweise — 668	585
Versionshinweise — 667	592
Versionshinweise — 666	596
Versionshinweise — 665	600
Versionshinweise — 664	603
Versionshinweise — 663	607
Versionshinweise — 662	610
Versionshinweise — 661	615
Versionshinweise — 660	619
Versionshinweise — 659	623
Versionshinweise — 658	628
Versionshinweise — 657	633
Versionshinweise — 656	637
Versionshinweise — 655	641
Versionshinweise — 654	644
Versionshinweise — 653	647

Versionshinweise — 652	649
Versionshinweise — 651	652
Versionshinweise — 650	654
Versionshinweise — 649	656
Versionshinweise — 648	659
Versionshinweise — 647	660
Versionshinweise — 646	662
Versionshinweise — 645	664
Versionshinweise — 644	665
Versionshinweise — 642	668
Versionshinweise — 641	669
Versionshinweise — 640	670
Version 1.0.640 Oracle-Änderungen	670
Version 1.0.640 Änderungen an Microsoft SQL Server	677
Änderungen an MySQL-Version 1.0.640	682
Version 1.0.640 PostgreSQL-Änderungen	683
Änderungen an der Db2-LUW-Version 1.0.640	686
Teradata-Änderungen an Version 1.0.640	686
Version 1.0.640 ändert sich für andere Engines	688
Dokumentverlauf	691
Frühere Aktualisierungen	708
.....	dccxviii

Was ist die AWS Schema Conversion Tool?

Mit dem AWS Schema Conversion Tool (AWS SCT) können Sie Ihr vorhandenes Datenbankschema von einer Datenbank-Engine in eine andere konvertieren. Sie können relationale OLTP-Schemata sowie Data Warehouse-Schemata konvertieren. Ihr konvertiertes Schema ist für Amazon Relational Database Service (Amazon RDS) MySQL, MariaDB, Oracle, SQL Server, PostgreSQL DB, einen Amazon-Aurora-DB-Cluster oder einen Amazon-Redshift-Cluster geeignet. Das konvertierte Schema kann auch für eine Datenbank in einer Amazon EC2-Instance verwendet oder in Form von Daten als Amazon S3-Bucket gespeichert werden.

AWS SCT unterstützt verschiedene Industriestandards, darunter Federal Information Processing Standards (FIPS), für Verbindungen zu einem Amazon S3-Bucket oder einer anderen AWS Ressource. AWS SCT entspricht auch dem Federal Risk and Authorization Management Program (FedRAMP). Weitere Informationen zu AWS und Compliance-Bemühungen finden Sie unter [Im Rahmen des Compliance-Programms zugelassene AWS-Services](#).

AWS SCT unterstützt folgende OLTP-Konvertierungen.

Quelldatenbank	Zieldatenbank
IBM Db2 für z/OS (Version 12)	Amazon Aurora MySQL-kompatible Ausgabe (Aurora MySQL), Amazon Aurora PostgreSQL-kompatible Ausgabe (Aurora PostgreSQL), MySQL, PostgreSQL Weitere Informationen finden Sie unter IBM Db2 for z/OS als Quelle verwenden .
IBM Db2 LUW (Versionen 9.1, 9.5, 9.7, 10.5, 11.1 und 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL Weitere Informationen finden Sie unter Verwenden von IBM Db2 LUW als Quelle .
Microsoft Azure SQL-Datenbank	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL

Quelldatenbank	Zieldatenbank
	Weitere Informationen finden Sie unter Azure SQL Database als Quelle verwenden .
Microsoft SQL Server (Version 2008 R2, 2012, 2014, 2016, 2017, 2019 und 2022)	Aurora MySQL, Aurora PostgreSQL, Babelfish für Aurora PostgreSQL (nur für Bewertung sberichte), MariaDB, Microsoft SQL Server, MySQL, PostgreSQL Weitere Informationen finden Sie unter Verwenden von SQL Server als Quelle .
MySQL (Version 5.5 und höher)	Aurora PostgreSQL, MySQL, PostgreSQL Weitere Informationen finden Sie unter Verwenden von MySQL als Quelle . Sie können Schema und Daten von MySQL zu einem Aurora MySQL-DB-Cluster migrieren , ohne es zu verwendenAWS SCT. Weitere Informationen finden Sie unter Migrieren von Daten zu einem Amazon Aurora-DB-Cluster .
Oracle (Version 10.1 und höher)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL Weitere Informationen finden Sie unter Oracle Database als Quelle verwenden .
PostgreSQL (Version 9.1 und höher)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL Weitere Informationen finden Sie unter Verwenden von PostgreSQL als Quelle .

Quelldatenbank	Zieldatenbank
SAP ASE (Versionen 12.5.4, 15.0.2, 15.5, 15.7 und 16.0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL Weitere Informationen finden Sie unter SAP ASE (Sybase ASE) als Quelle verwenden .

AWS SCT unterstützt die folgenden Data Warehouse-Konvertierungen.

Quell-Data-Warehouse	Ziel-Data Warehouse
Amazon Redshift	Amazon Redshift Weitere Informationen finden Sie unter Amazon Redshift als Quelle verwenden .
Azure Synapse Analytics	Amazon Redshift Weitere Informationen finden Sie unter Azure Synapse Analytics als Quelle verwenden .
BigQuery	Amazon Redshift Weitere Informationen finden Sie unter Verwenden von BigQuery als Quelle .
Greenplum Database (Versionen 4.3 und 6.21)	Amazon Redshift Weitere Informationen finden Sie unter Greenplum Database als Quelle verwenden .
Microsoft SQL Server (Version 2008 und höher)	Amazon Redshift Weitere Informationen finden Sie unter SQL Server Data Warehouse als Quelle verwenden .
Netezza (Version 7.0.3 und höher)	Amazon Redshift

Quell-Data-Warehouse	Ziel-Data Warehouse
	Weitere Informationen finden Sie unter Netezza als Quelle verwenden .
Oracle (Version 10.1 und höher)	Amazon Redshift Weitere Informationen finden Sie unter Oracle Data Warehouse als Quelle verwenden .
Schneeflocke (Version 3)	Amazon Redshift Weitere Informationen finden Sie unter Snowflake als Quelle verwenden .
Teradata (Version 13 und höher)	Amazon Redshift Weitere Informationen finden Sie unter Teradata als Quelle verwenden .
Vertica (Version 7.2.2 und höher)	Amazon Redshift Weitere Informationen finden Sie unter Vertica als Quelle verwenden .

AWS SCT unterstützt die folgenden NoSQL-Datenbankkonvertierungen von Daten.

Quelldatenbank	Zieldatenbank
Apache Cassandra (Versionen 2.1.x, 2.2.16 und 3.11.x)	Amazon DynamoDB Weitere Informationen finden Sie unter Apache Cassandra als Quelle verwenden .

AWS SCT unterstützt Konvertierungen der folgenden ETL-Prozesse (Extrahieren, Transformieren und Laden). Weitere Informationen finden Sie unter [ETL-Prozesse konvertieren](#).

Quelle	Ziel
Informatica ETL-Skripts	Informatica
ETL-Pakete für Microsoft SQL Server Integration Services (SSIS)	AWS Glue oder AWS Glue Studio
Shell-Skripte mit eingebetteten Befehlen aus Teradata Basic Teradata Query (BTEQ)	Amazon-Redshift-RSQL
Teradata BTEQ ETL-Skripts	AWS Glue oder Amazon Redshift RSQL
FastExportTeradata-Jobskripte	Amazon-Redshift-RSQL
FastLoadTeradata-Jobskripte	Amazon-Redshift-RSQL
MultiLoadTeradata-Jobskripte	Amazon-Redshift-RSQL

AWS SCT unterstützt die folgenden Big-Data-Framework-Migrationen. Weitere Informationen finden Sie unter [Migration von Big-Data-Frameworks](#).

Quelle	Ziel
Apache Hive (Version 0.13.0 und höher)	Hive auf Amazon EMR
Apache HDFS	Amazon S3 oder HDFS auf Amazon EMR
Apache Oozie	AWS Step Functions

Übersicht über die Schemakonvertierung

AWS SCT bietet eine projektbasierte Benutzeroberfläche, um das Datenbankschema Ihrer Quelldatenbank automatisch in ein Format zu konvertieren, das mit Ihrer Amazon RDS-Zielinstanz kompatibel ist. Wenn ein Schema aus Ihrer Quelldatenbank nicht automatisch konvertiert werden kann, finden Sie AWS SCT hier eine Anleitung, wie Sie ein gleichwertiges Schema in Ihrer Amazon RDS-Zieldatenbank erstellen können.

Informationen zur Installation von AWS SCT finden Sie unter [Installation, Überprüfung und Aktualisierung AWS SCT](#).

Eine Einführung in die AWS SCT-Benutzeroberfläche finden Sie unter [Verwendung der AWS SCT Benutzeroberfläche](#).

Weitere Informationen zum Umwandlungsverfahren finden Sie unter [Konvertieren von Datenbankschemas mithilfe von AWS SCT](#).

Neben der Konvertierung Ihres vorhandenen Datenbankschemas von einer Datenbank-Engine in eine andere AWS SCT bietet es einige zusätzliche Funktionen, mit denen Sie Ihre Daten und Anwendungen in die AWS Cloud verschieben können:

- Sie können Datenextraktionsagenten verwenden, um Daten aus Ihrem Data Warehouse zu extrahieren, um die Migration zu Amazon Redshift vorzubereiten. Zur Verwaltung der Agenten für die Datenextraktion können Sie das AWS SCT verwenden. Weitere Informationen finden Sie unter [Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift](#).
- Erstellen Sie mit dem AWS SCT AWS DMS-Endpunkte und -Aufgaben. Sie können diese Aufgaben mit dem AWS SCT ausführen und überwachen. Weitere Informationen finden Sie unter [Verwenden von AWS SCT mit AWS DMS](#).
- In einigen Fällen können Datenbankfunktionen nicht in gleichwertige Amazon RDS- oder Amazon Redshift-Funktionen umgewandelt werden. Mit dem Assistenten des AWS SCT-Erweiterungspaketes können Sie AWS Lambda-Funktionen und Python-Bibliotheken installieren, um diejenigen Funktionen zu emulieren, die nicht konvertiert werden können. Weitere Informationen finden Sie unter [Verwenden von AWS SCT Erweiterungspaket](#).
- Sie können es verwenden AWS SCT, um Ihre bestehende Amazon Redshift-Datenbank zu optimieren. AWS SCT empfiehlt Sortier- und Verteilungsschlüssel, um Ihre Datenbank zu optimieren. Weitere Informationen finden Sie unter [Optimierung von Amazon Redshift mithilfe von AWS SCT](#).
- Sie können AWS SCT es verwenden, um Ihr vorhandenes lokales Datenbankschema in eine Amazon RDS-DB-Instance zu kopieren, auf der dieselbe Engine ausgeführt wird. Mit dieser Funktion können Sie potenzielle Kosteneinsparungen kalkulieren, die beim Wechsel in die Cloud und einer Änderung des Lizenztyps möglich sind.
- Sie können mit dem AWS SCT SQL-Code in Ihren C++-, C #-, Java- oder anderen Anwendungscode konvertieren. Sie können den konvertierten SQL-Code anzeigen, analysieren, bearbeiten und speichern. Weitere Informationen finden Sie unter [Anwendungs-SQL konvertieren mit AWS SCT](#).

- Sie können AWS SCT verwenden, um Extraktions-, Transformations- und Ladeprozesse (ETL) zu migrieren. Weitere Informationen finden Sie unter [Konvertierung von Extrahieren, Transformieren und Laden \(ETL\) -Prozessen mit AWS Schema Conversion Tool](#).

Feedback geben

Sie können Feedback geben zu AWS SCT. Sie können einen Fehlerbericht einreichen, eine Funktionsanfrage einreichen oder allgemeine Informationen angeben.

Um Feedback zu geben über AWS SCT

1. Starten Sie den AWS Schema Conversion Tool.
2. Öffnen Sie das Menü Help und klicken Sie auf Leave Feedback. Das Dialogfeld Leave Feedback wird angezeigt.
3. Wählen Sie unter Area eine der Optionen Information, Bug report oder Feature request aus.
4. Wählen Sie für Source database Ihre Quelldatenbank aus. Wählen Sie Any aus, wenn Ihr Feedback sich nicht auf eine bestimmte Datenbank bezieht.
5. Wählen Sie für Target database Ihre Zieldatenbank aus. Wählen Sie Any aus, wenn Ihr Feedback sich nicht auf eine bestimmte Datenbank bezieht.
6. Geben Sie unter Title einen Titel für Ihr Feedback ein.
7. Geben Sie unter Message Ihr Feedback ein.
8. Klicken Sie auf Send, um das Feedback abzugeben.

Installation, Überprüfung und Aktualisierung AWS SCT

The AWS Schema Conversion Tool (AWS SCT) ist eine eigenständige Anwendung, die eine projektbasierte Benutzeroberfläche bietet. AWS SCT ist für Microsoft Windows, Fedora Linux und Ubuntu Linux verfügbar. AWS SCT wird nur auf 64-Bit-Betriebssystemen unterstützt.

Um sicherzustellen, dass Sie die richtige Version der AWS SCT Distributionsdatei erhalten, führen wir nach dem Herunterladen der komprimierten Datei Schritte zur Überprüfung durch. Sie können die Datei unter Verwendung der bereitgestellten Schritte überprüfen.

AWS SCT ist sowohl als eigenständige Anwendung als auch als Befehlszeilentool verfügbar. Hinweise zum Befehlszeilentool finden Sie unter [AWS SCT CLI](#)

Themen

- [Wird installiert AWS SCT](#)
- [Der AWS SCT Dateidownload wird überprüft](#)
- [Die erforderlichen Datenbanktreiber werden heruntergeladen](#)
- [Aktualisierung AWS SCT](#)
- [AWS SCT CLI](#)

Wird installiert AWS SCT

Sie können AWS SCT auf den folgenden Betriebssystemen installieren:

- Microsoft Windows 10
- Fedora Linux 3.6 und höher
- Ubuntu Linux 18 und höher

Um zu installieren AWS SCT

1. Laden Sie die komprimierte Datei herunter, die das AWS SCT Installationsprogramm enthält, indem Sie den Link für Ihr Betriebssystem verwenden. Alle komprimierten Dateien haben eine ZIP-Erweiterung. Wenn Sie die AWS SCT Installationsdatei extrahieren, wird sie das für Ihr Betriebssystem geeignete Format haben.

- [Microsoft Windows](#)
- [Ubuntu Linux \(.deb\)](#)
- [Fedora Linux \(.rpm\)](#)

2. Extrahieren Sie die AWS SCT Installationsdatei für Ihr Betriebssystem, wie unten gezeigt.

Betriebssystem	Dateiname
Fedora Linux	aws-schema-conversion-tool-1.0. <i>build-number</i> .x86_64.rpm
Microsoft Windows	AWS Schema Conversion Tool-1.0. <i>build-number</i> .msi
Ubuntu Linux	aws-schema-conversion-tool-1.0. <i>build-number</i> .deb

3. Führen Sie die im vorherigen Schritt extrahierte AWS SCT Installationsdatei aus. Befolgen Sie die Anleitung für Ihr Betriebssystem:

Betriebssystem	Anweisungen installieren
Fedora Linux	Führen Sie in dem Verzeichnis, in das Sie die Datei heruntergeladen haben, den folgenden Befehl aus: <pre>sudo yum install aws-schema-conversion-tool-1.0. <i>build-number</i> .x86_64.rpm</pre>
Microsoft Windows	Doppelklicken Sie auf die Datei, um das Installationsprogramm auszuführen.
Ubuntu Linux	Führen Sie in dem Verzeichnis, in das Sie die Datei heruntergeladen haben, den folgenden Befehl aus: <pre>sudo dpkg -i aws-schema-conversion-tool-1.0. <i>build-number</i> .deb</pre>

4. Laden Sie die Java Database Connectivity (JDBC) -Treiber für Ihre Quell- und Zieldatenbank-Engines herunter. Informationen und Download-Links finden Sie unter [Die erforderlichen Datenbanktreiber werden heruntergeladen.](#)

Jetzt haben Sie die Einrichtung der AWS SCT Anwendung abgeschlossen. Doppelklicken Sie auf das Anwendungssymbol, um es auszuführen AWS SCT.

Der AWS SCT Dateidownload wird überprüft

Es gibt verschiedene Möglichkeiten, die Distributionsdatei von AWS SCT zu überprüfen. Am einfachsten ist es, die Prüfsumme der Datei mit der veröffentlichten Prüfsumme von zu vergleichen. AWS Als zusätzliche Sicherheitsstufe können Sie mithilfe der folgenden Verfahren die Distributionsdatei basierend auf dem Betriebssystem, auf dem die Datei installiert ist, überprüfen.

In diesem Abschnitt werden folgende Themen behandelt.

Themen

- [Überprüfung der Prüfsumme der Datei AWS SCT](#)
- [Überprüfung der AWS SCT RPM-Dateien auf Fedora](#)
- [Überprüfung der AWS SCT DEB-Dateien auf Ubuntu](#)
- [Überprüfen der AWS SCT MSI-Datei unter Microsoft Windows](#)

Überprüfung der Prüfsumme der Datei AWS SCT

Um Fehler zu erkennen, die beim Herunterladen oder Speichern der AWS SCT komprimierten Datei entstanden sein könnten, können Sie die Dateiprüfsumme mit einem Wert vergleichen, der von bereitgestellt wird. AWS AWS verwendet den SHA256-Algorithmus für die Prüfsumme.

Um die AWS SCT Distributionsdatei anhand einer Prüfsumme zu überprüfen

1. Laden Sie die AWS SCT Distributionsdatei über die Links im Abschnitt Installation herunter. Weitere Informationen finden Sie unter [Wird installiert AWS SCT](#).
2. Laden Sie die neueste Prüfsummendatei mit dem Namen [sha256Check.txt](#) herunter. Diese Datei enthält die Prüfsummen für die neueste AWS SCT Version. Die Datei kann beispielsweise wie folgt aussehen:

```
Fedora    b4f5f66f91bfcc1b312e2827e960691c269a9002cd1371cf1841593f88cbb5e6
Ubuntu    4315eb666449d4fcd95932351f00399adb6c6cf64b9f30adda2eec903c54eca4
Windows   6e29679a3c53c5396a06d8d50f308981e4ec34bd0acd608874470700a0ae9a23
```

- Führen Sie den SHA256-Validierungsbefehl für Ihr Betriebssystem in dem Verzeichnis mit der Distributionsdatei aus. Führen Sie beispielsweise den folgenden Befehl unter Linux aus.

```
shasum -a 256 aws-schema-conversion-tool-1.0.latest.zip
```

- Vergleichen Sie die Ergebnisse des Befehls mit dem Wert in der Datei sha256Check.txt. Wenn die Prüfsummen übereinstimmen, ist es sicher, die Distributionsdatei auszuführen. Wenn die Prüfsummen nicht übereinstimmen, führen Sie die Verteilungsdatei nicht aus und [wenden Sie sich an den AWS-Support](#).

Überprüfung der AWS SCT RPM-Dateien auf Fedora

AWS bietet zusätzlich zur Prüfsumme der Distributionsdatei eine weitere Validierungsebene. Alle RPM-Dateien in der Distributionsdatei sind mit einem AWS privaten Schlüssel signiert. Sie finden den öffentlichen GPG-Schlüssel unter [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key).

Um die AWS SCT RPM-Dateien auf Fedora zu überprüfen

- Laden Sie die AWS SCT Distributionsdatei über die Links im Abschnitt Installation herunter.
- Überprüfen Sie die Prüfsumme der AWS SCT Distributionsdatei.
- Extrahieren Sie den Inhalt der Distributionsdatei. Suchen Sie die RPM-Datei, die Sie überprüfen möchten.
- Laden Sie den öffentlichen GPG-Schlüssel unter [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key) herunter.
- Importieren Sie den öffentlichen Schlüssel in Ihre RPM-DB (stellen Sie sicher, dass Sie über die entsprechenden Berechtigungen verfügen), indem Sie den folgenden Befehl ausführen:

```
sudo rpm --import aws-dms-team@amazon.com.public.gpg-key
```

- Überprüfen Sie, ob der Import erfolgreich war, indem Sie den folgenden Befehl ausführen:

```
rpm -q --qf "%{NAME}-%{VERSION}-%{RELEASE} \n %{SUMMARY} \n" gpg-pubkey-  
ea22abf4-5a21d30c
```

- Überprüfen Sie die RPM-Signatur, indem Sie den folgenden Befehl ausführen:

```
rpm --checksig -v aws-schema-conversion-tool-1.0.build number-1.x86_64.rpm
```


Überprüfung der AWS SCT DEB-Dateien auf Ubuntu

AWS bietet zusätzlich zur Prüfsumme der Distributionsdatei eine weitere Validierungsebene. Alle DEB-Dateien in der Distributionsdatei werden mit einer von GPG getrennten Signatur signiert.

Um die AWS SCT DEB-Dateien auf Ubuntu zu überprüfen

1. Laden Sie die AWS SCT Distributionsdatei über die Links im Abschnitt Installation herunter.
2. Überprüfung der Prüfsumme der AWS SCT Distributionsdatei.
3. Extrahieren Sie den Inhalt der Distributionsdatei. Suchen Sie die DEB-Datei, die Sie überprüfen möchten.
4. [Laden Sie die abgetrennte Signatur von aws-schema-conversion-tool -1.0.latest.deb.asc herunter.](#)
5. Laden Sie den öffentlichen GPG-Schlüssel unter [amazon.com.public.gpg-key](#) herunter.
6. Importieren Sie den öffentlichen GPG-Schlüssel, indem Sie den folgenden Befehl ausführen:

```
gpg --import aws-dms-team@amazon.com.public.gpg-key
```

7. Überprüfen Sie die Signatur, indem Sie den folgenden Befehl ausführen:

```
gpg --verify aws-schema-conversion-tool-1.0.latest.deb.asc aws-schema-conversion-tool-1.0.build number.deb
```

Überprüfen der AWS SCT MSI-Datei unter Microsoft Windows

AWS bietet zusätzlich zur Prüfsumme der Distributionsdatei eine weitere Überprüfungsebene. Die MSI-Datei hat eine digitale Signatur, mit der Sie überprüfen können, ob sie signiert AWS wurde.

Um die AWS SCT MSI-Datei unter Windows zu überprüfen

1. Laden Sie die AWS SCT Distributionsdatei über die Links im Abschnitt Installation herunter.
2. Überprüfung der Prüfsumme der AWS SCT Distributionsdatei.
3. Extrahieren Sie den Inhalt der Distributionsdatei. Suchen Sie die MSI-Datei, die Sie überprüfen möchten.
4. Klicken Sie im Windows Explorer mit der rechten Maustaste auf die MSI-Datei und wählen Sie Properties.

5. Wählen Sie die Registerkarte Digital Signatures aus.
6. Überprüfen Sie, ob die digitale Signatur von Amazon Services LLC stammt.

Die erforderlichen Datenbanktreiber werden heruntergeladen

Laden Sie AWS SCT die JDBC-Treiber für Ihre Quell- und Zieldatenbank-Engines herunter, damit sie ordnungsgemäß funktionieren. Wenn Sie eine virtuelle Zieldatenbankplattform verwenden, müssen Sie den JDBC-Treiber für Ihre Zieldatenbank-Engine nicht herunterladen. Weitere Informationen finden Sie unter [Verwendung virtueller Ziele](#).

Nachdem Sie die Treiber heruntergeladen haben, geben Sie den Speicherort der Treiberdateien an. Weitere Informationen finden Sie unter [Speichern von Treiberpfaden in den globalen Einstellungen](#).

Sie können die Datenbanktreiber an folgenden Speicherorten herunterladen.

Important

Laden Sie die neueste verfügbare Version des Treibers herunter. Die folgende Tabelle enthält die niedrigste Version des Datenbanktreibers, die von unterstützt wird AWS SCT.

Datenbank-Engine	Treiber	Speicherort für den Download
Amazon Aurora MySQL-Compatible Edition	mysql-connector-java-5.1.6.jar	https://www.mysql.com/products/connector/
Amazon Aurora PostgreSQL-Compatible Edition	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar

Datenbank-Engine	Treiber	Speicherort für den Download
Amazon EMR	HiveJDBC42.jar	http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/
Amazon-Redshift	redshift-jdbc42-2.1.0.9.jar	https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
Amazon Redshift Serverless	redshift-jdbc42-2.1.0.9.jar	https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
Apache Hive	hive-jdbc-2.3.4-standalone.jar	https://repo1.maven.org/maven2/org/apache/hive/hive-jdbc/2.3.4/hive-jdbc-2.3.4-standalone.jar
Azure SQL Database	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-Treiber?sql-server-veransicht=15#72
Azure Synapse-Analytik	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-Treiber?sql-server-veransicht=15#72
Greenplum-Datenbank	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
IBM Db2 für z/OS	db2jcc-db2jcc4.jar	https://www.ibm.com/support/pages/db2--downloads-db2-zos-jdbc-driver-versions-and
IBM Db2 (LUW)	db2jcc-db2jcc4.jar	https://www.ibm.com/support/pages/node/382667
MariaDB	mariadb-java-client-2.4.1.jar	https://downloads.mariadb.com/Connectors/java/connector-java-2.4.1/mariadb-java-client-2.4.1.jar

Datenbank-Engine	Treiber	Speicherort für den Download
Microsoft SQL Server	mssql-jdbc-10.2.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-?for-sql-server&view=15sql-server-ver
MySQL	mysql-connector-java-8.0.15.jar	https://dev.mysql.com/downloads/connector/j/
Netezza	nzjdbc.jar Verwenden Sie die Client-Tools-Software. Laden Sie die Treiberversion 7.2.1 herunter, die abwärtskompatibel mit Data Warehouse-Version 7.2.0 ist.	http://www.ibm.com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.datacon.doc/c_datacon_plg_overview.html
Oracle	ojdbc8.jar Die Treiberversionen 8 und höher werden unterstützt.	https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html
PostgreSQL	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
SAP ASE (Sybase ASE)	jconn4.jar	Der jConnect JDBC-Treiber
Snowflake	snowflake-jdbc-3.9.2.jar Weitere Informationen finden Sie unter Den JDBC-Treiber herunterladen/integrieren .	https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9.2.jar

Datenbank-Engine	Treiber	Speicherort für den Download
Teradata	terajdbc4.jar tdgssconfig.jar Für Teradata JDBC-Treiberversion 16.20.00.11 und höher benötigen Sie die Datei nicht. tdgssconfig.jar	https://downloads.teradata.com/download/connectivity/jdbc-driver
Vertica	vertica-jdbc-9.1.1-0.jar Treiberversionen 7.2.0 und höher werden unterstützt.	https://www.vertica.com/client_drivers/9.1.x/9.1.1-0/vertica-jdbc-9.1.1-0.jar

Installation von JDBC-Treibern unter Linux

Sie können die folgenden Schritte ausführen, um die JDBC-Treiber auf Ihrem Linux-System zur Verwendung mit zu installieren. AWS SCT

So installieren Sie die JDBC-Treiber auf Ihrem Linux-System

1. Erstellen Sie ein Verzeichnis, in dem Sie die JDBC-Treiber speichern möchten.

```
PROMPT>sudo mkdir -p /usr/local/jdbc-drivers
```

2. Installieren Sie die JDBC-Treiber für Ihre Datenbank-Engine mit den folgenden Befehlen.

Datenbank-Engine	Befehle zur Installation
Amazon Aurora (kompatibel mit MySQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>

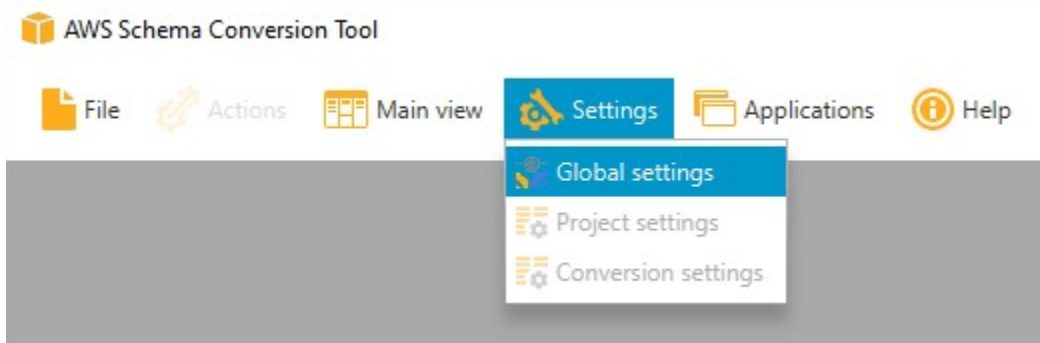
Datenbank-Engine	Befehle zur Installation
Amazon Aurora (kompatibel mit PostgreSQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>
Microsoft SQL Server	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/sqljdbc_X.X.X_enu.tar.gz</pre>
MySQL	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>
Oracle	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo mkdir oracle-jdbc PROMPT> cd oracle-jdbc PROMPT> sudo cp -a /tmp/ojdbc8.jar .</pre>
PostgreSQL	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>

Speichern von Treiberpfaden in den globalen Einstellungen

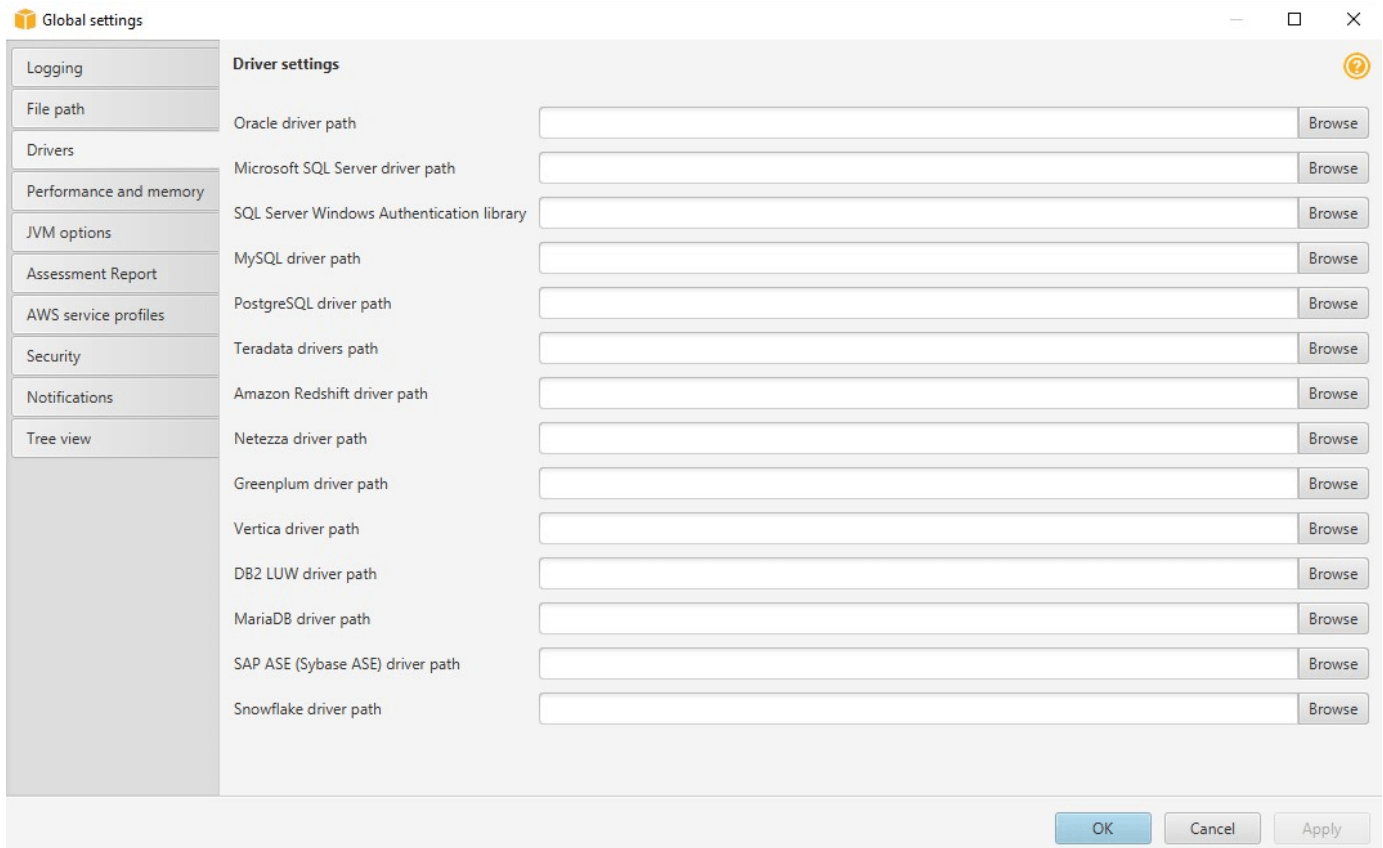
Nachdem Sie die erforderlichen JDBC-Treiber heruntergeladen und installiert haben, können Sie den Speicherort der Treiber global in den AWS SCT Einstellungen festlegen. Wenn Sie den Speicherort der Treiber nicht global festlegen, müssen Sie diesen bei jedem Verbindungsvorgang zu einer Datenbank in der Anwendung angeben.

So aktualisieren Sie den Speicherort der Treiberdatei

1. Wählen Sie in AWS SCT Einstellungen und anschließend Allgemeine Einstellungen aus.



2. Klicken Sie unter Global Settings auf Drivers. Fügen Sie den Dateipfad zum JDBC-Treiber für Ihre Quelldatenbank-Engine und Ihre Amazon RDS-DB-Instance-Ziel-Datenbank-Engine hinzu.



3. Wenn Sie die Treiberpfade hinzugefügt haben, klicken Sie auf OK.

Aktualisierung AWS SCT

AWS regelmäßige Updates AWS SCT mit neuen Features und Funktionen. Wenn Sie von einer früheren Version aktualisieren, erstellen Sie ein neues AWS SCT Projekt und konvertieren Sie alle Datenbankobjekte, die Sie verwenden, erneut.

Sie können überprüfen, ob Updates für AWS SCT vorhanden sind.

Um nach Updates zu suchen AWS SCT

1. Wenn Sie angemeldet AWS SCT sind, wählen Sie „Hilfe“ und anschließend „Nach Updates suchen“.
2. Wählen Sie im Dialogfeld Check for Updates (Nach Updates suchen) die Option What's New (Neues) aus. Wenn der Link nicht angezeigt wird, haben Sie die neueste Version.

AWS SCT CLI

Sie können die AWS SCT CLI für die Befehlszeilenverwendung herunterladen. Verwenden Sie den folgenden Link, um das JAR herunterzuladen:

[AWSSchemaConversionToolBatch.jar](#)

Verwendung der AWS SCT Benutzeroberfläche

Verwenden Sie die folgenden Themen, die Ihnen beim Arbeiten mit der AWS SCT Benutzeroberfläche helfen. Informationen zur Installation AWS SCT finden Sie unter [Installation, Überprüfung und Aktualisierung AWS SCT](#).

Themen

- [Das AWS SCT Projektfenster](#)
- [Starten von AWS SCT](#)
- [Ein AWS SCT Projekt erstellen](#)
- [Verwenden eines Assistenten für neue Projekte in AWS SCT](#)
- [AWS SCT Projekt speichern und öffnen](#)
- [Hinzufügen von Datenbankservern zu einem AWS SCT Projekt](#)
- [AWS SCT Im Offline-Modus ausführen](#)
- [AWS SCT Baumfilter verwenden](#)
- [Schemas in der AWS SCT Baumansicht ausblenden](#)
- [Erstellung und Überprüfung des Bewertungsberichts zur Datenbankmigration](#)
- [Konvertieren Ihres Schemas](#)
- [Anwenden des konvertierten Schemas auf Ihre Ziel-DB-Instance](#)
- [Speichern von AWS Serviceprofilen in AWS SCT](#)
- [Verwenden von AWS Secrets Manager](#)
- [Speichern von Datenbankpasswörtern](#)
- [Verwenden der UNION ALL-Ansicht für Projekte mit partitionierten Tabellen](#)
- [Tastenkombinationen für AWS SCT](#)

Das AWS SCT Projektfenster

Die folgende Abbildung zeigt, was Sie sehen, AWS SCT wenn Sie ein Schemamigrationsprojekt erstellen und anschließend ein Schema konvertieren.

1. Im linken Bereich wird das Schema der Quelldatenbank in einer Strukturansicht präsentiert. Ihr Datenbankschema ist "lazy loaded". Anders ausgedrückt, wenn Sie ein Element in der

- Strukturansicht auswählen, ruft AWS SCT das aktuelle Schema aus Ihrer Quelldatenbank ab und zeigt dieses an.
- Im oberen mittleren Bereich werden Aktionselemente für Schemaelemente aus der Quelldatenbank-Engine angezeigt, die nicht automatisch in die Zieldatenbank-Engine konvertiert werden konnten.
 - Im rechten Bereich wird das Schema der Ziel-DB-Instance in einer Strukturansicht präsentiert. Ihr Datenbankschema ist "lazy loaded". Wenn Sie an dieser Stelle ein Element aus der Strukturansicht auswählen, ruft AWS SCT das aktuelle Schema aus Ihrer Zieldatenbank ab und zeigt dieses an.

The screenshot shows the AWS Schema Conversion Tool interface. The left sidebar (1) displays a tree view of the source Microsoft SQL Server database structure. The central pane (2) lists conversion issues, with the third issue highlighted in blue. The right sidebar (3) shows the target Amazon RDS for MySQL database structure. The bottom section (4) displays a comparison of the 'Address' table between the source and target, including a table with properties and a SQL DDL snippet (5).

1

2

3

4

5

Properties	SQL
Created or last modified	
Created	2016-01-05 23:29:41.09
Last modified	2016-01-05 23:46:59.76
Object name	
Name	Address
object-id	693577509
schema-id	33
type	U
type-desc	USER_TABLE
Table extended properties	
is-memory-optimized	N

```

CREATE TABLE IF NOT EXISTS SampleDatabase_Customers
(
  AddressID INT NOT NULL,
  AddressLine1 VARCHAR(60) NOT NULL,
  AddressLine2 VARCHAR(60) DEFAULT NULL,
  City VARCHAR(30) NOT NULL,
  StateProvinceID INT NOT NULL,
  PostalCode VARCHAR(15) NOT NULL,
  SpatialLocation VARCHAR(8000) DEFAULT NULL,
  rowguid CHAR(38) DEFAULT NULL,
  ModifiedDate DATETIME DEFAULT NULL,
  CONSTRAINT PK_Address PRIMARY KEY (AddressID)
) ENGINE=InnoDB

```

4. Wenn Sie im unteren linken Bereich ein Schemaelement auswählen, werden Eigenschaften angezeigt. Diese beschreiben das Quellschemaelement und den SQL-Befehl zum Erstellen dieses Elements in der Quelldatenbank.
5. Wenn Sie im unteren rechten Bereich ein Schemaelement auswählen, werden Eigenschaften angezeigt. Diese beschreiben das Zielschemaelement und den SQL-Befehl zum Erstellen dieses Elements in der Zieldatenbank. Sie können diesen SQL-Befehl bearbeiten und die aktualisierte Version in Ihrem Projekt speichern.

Starten von AWS SCT

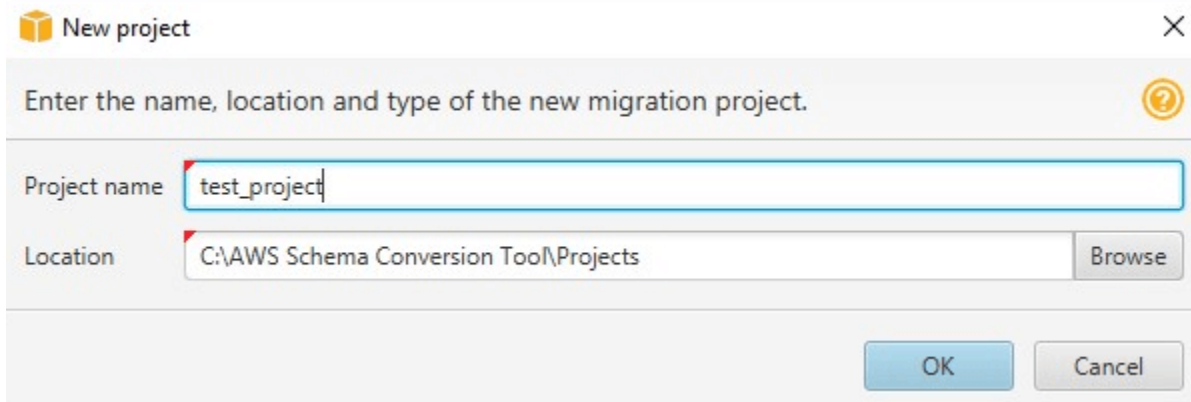
Doppelklicken Sie auf das Anwendungssymbol AWS Schema Conversion Tool, um die zu starten.

Ein AWS SCT Projekt erstellen

Gehen Sie wie folgt vor, um ein AWS Schema Conversion Tool Projekt zu erstellen.

So erstellen Sie Ihr Projekt

1. Starten Sie den AWS Schema Conversion Tool.
2. Wählen Sie im Menü Datei die Option Neues Projekt aus. Das Dialogfeld Neues Projekt wird angezeigt.



3. Geben Sie einen Namen für Ihr Projekt ein, das lokal auf Ihrem Computer gespeichert wird.
4. Geben Sie den Speicherort für Ihre lokale Projektdatei an.
5. Wählen Sie OK, um Ihr AWS SCT-Projekt zu erstellen.
6. Wählen Sie Quelle hinzufügen, um Ihrem AWS SCT Projekt eine neue Quelldatenbank hinzuzufügen. Sie können Ihrem AWS SCT Projekt mehrere Quelldatenbanken hinzufügen.

7. Wählen Sie Ziel hinzufügen, um Ihrem AWS SCT Projekt eine neue Zielplattform hinzuzufügen. Sie können Ihrem AWS SCT Projekt mehrere Zielplattformen hinzufügen.
8. Wählen Sie das Quelldatenbankschema im linken Bereich aus.
9. Geben Sie im rechten Bereich die Zieldatenbankplattform für das ausgewählte Quellschema an.
10. Wählen Sie Create Mapping. Diese Schaltfläche wird aktiv, nachdem Sie das Quelldatenbankschema und die Zieldatenbankplattform ausgewählt haben. Weitere Informationen finden Sie unter [Mapping-Regeln erstellen](#).

Jetzt ist Ihr AWS SCT Projekt eingerichtet. Sie können Ihr Projekt speichern, einen Bewertungsbericht zur Datenbankmigration erstellen und Ihre Quelldatenbankschemas konvertieren.

Verwenden eines Assistenten für neue Projekte in AWS SCT

Mit dem Assistenten für neue Projekte können Sie ein neues Datenbankmigrationsprojekt erstellen. Dieser Assistent unterstützt Sie bei der Festlegung Ihres Migrationsziels und beim Herstellen einer Verbindung zu Ihren Datenbanken. Es schätzt, wie komplex eine Migration für alle unterstützten Zielziele sein könnte. Nachdem Sie den Assistenten ausgeführt haben, wird ein zusammenfassender Bericht für die Migration Ihrer Datenbank zu verschiedenen Zielzielen in AWS SCT erstellt. Sie können diesen Bericht verwenden, um mögliche Zielziele zu vergleichen und den optimalen Migrationspfad auszuwählen.

Starten Sie den Assistenten für neue Projekte

1. Wählen Sie Ihre Quelldatenbank aus.
 - a. Starten Sie den AWS Schema Conversion Tool.
 - b. Wählen Sie im Menü Datei die Option Neuer Projektassistent aus. Das Dialogfeld Neues Datenbankmigrationsprojekt erstellen wird geöffnet.
 - c. Gehen Sie folgendermaßen vor, um die Verbindungsinformationen der Quelldatenbank einzugeben:

Parameter	Action
Project name	Geben Sie einen Namen für Ihr Projekt ein, das lokal auf Ihrem Computer gespeichert wird.

Parameter	Action
Ort	Geben Sie den Speicherort für Ihre lokale Projektdatei an.
Source type (Quellentyp)	<p>Wählen Sie eine der folgenden Optionen: SQL-Datenbank, NoSQL-Datenbank oder ETL.</p> <p>Wenn Sie den zusammenfassenden Bericht sehen möchten, der alle Migrationsziele enthält, wählen Sie SQL-Datenbank.</p>
Source engine (Quell-Engine)	Wählen Sie Ihre Quelldatenbank-Engine aus.
Strategie der Migration	<p>Wählen Sie eine der folgenden Optionen:</p> <ul style="list-style-type: none"> • Ich möchte die Engine wechseln und für die Cloud optimieren — Diese Option konvertiert Ihre Quelldatenbank in eine neue Datenbank-Engine. • Ich möchte dieselbe Engine behalten, aber für die Cloud optimieren — Mit dieser Option bleibt Ihre Datenbank-Engine unverändert und die Datenbank wird von der lokalen in die Cloud verschoben. • Ich möchte einen kombinierten Bericht für den Wechsel der Datenbank-Engine und die Optimierung für die Cloud sehen. Mit dieser Option wird die Migrationskomplexität aller verfügbaren Migrationsoptionen verglichen. <p>Wenn Sie den aggregierten Bewertungsbericht sehen möchten, der alle Migrationsziele enthält, wählen Sie die letzte Option.</p>

d. Wählen Sie Next (Weiter). Die Seite Mit der Quelldatenbank Connect wird geöffnet.

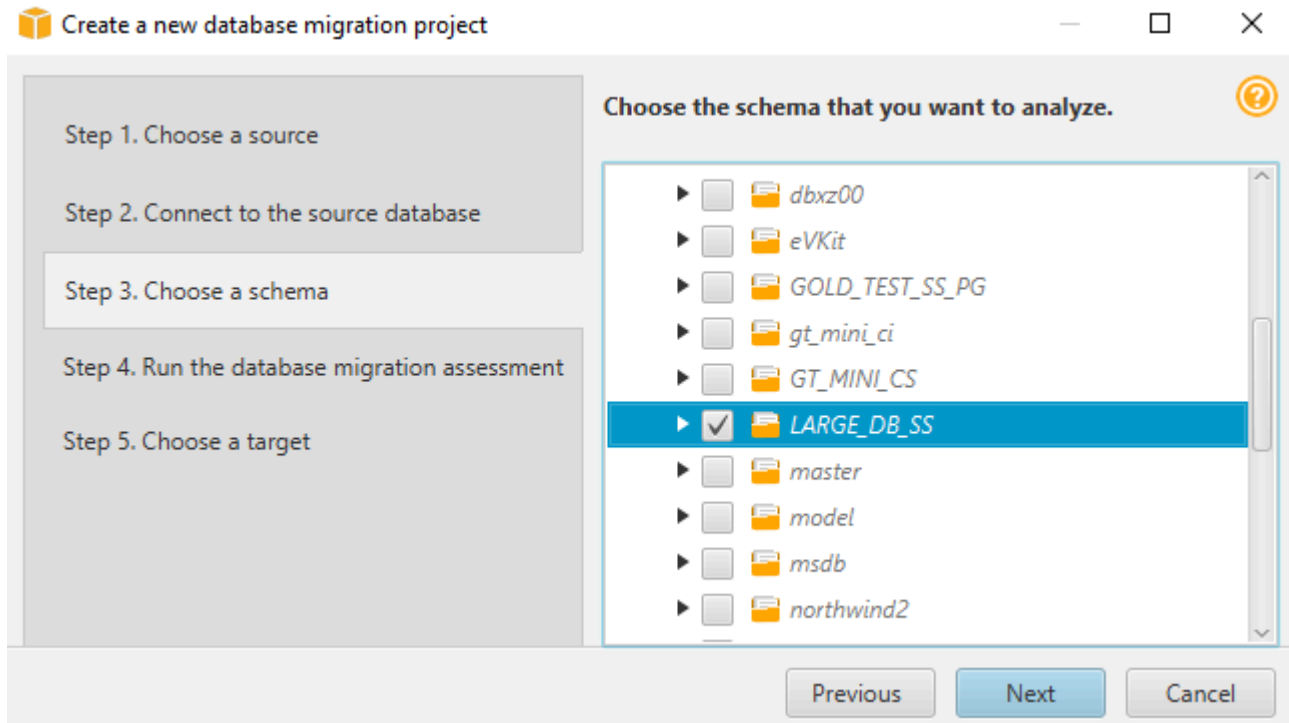
2. Stellen Sie eine Verbindung zur Quelldatenbank her.

a. Geben Sie Ihre Verbindungsinformationen für die Quelldatenbank an. Die Verbindungsparameter hängen von Ihrer Quelldatenbank-Engine ab. Stellen Sie sicher, dass der Benutzer, den Sie für die Analyse Ihrer Quelldatenbank verwenden, über die entsprechenden Berechtigungen verfügt. Weitere Informationen finden Sie unter [Quellen für AWS SCT](#).

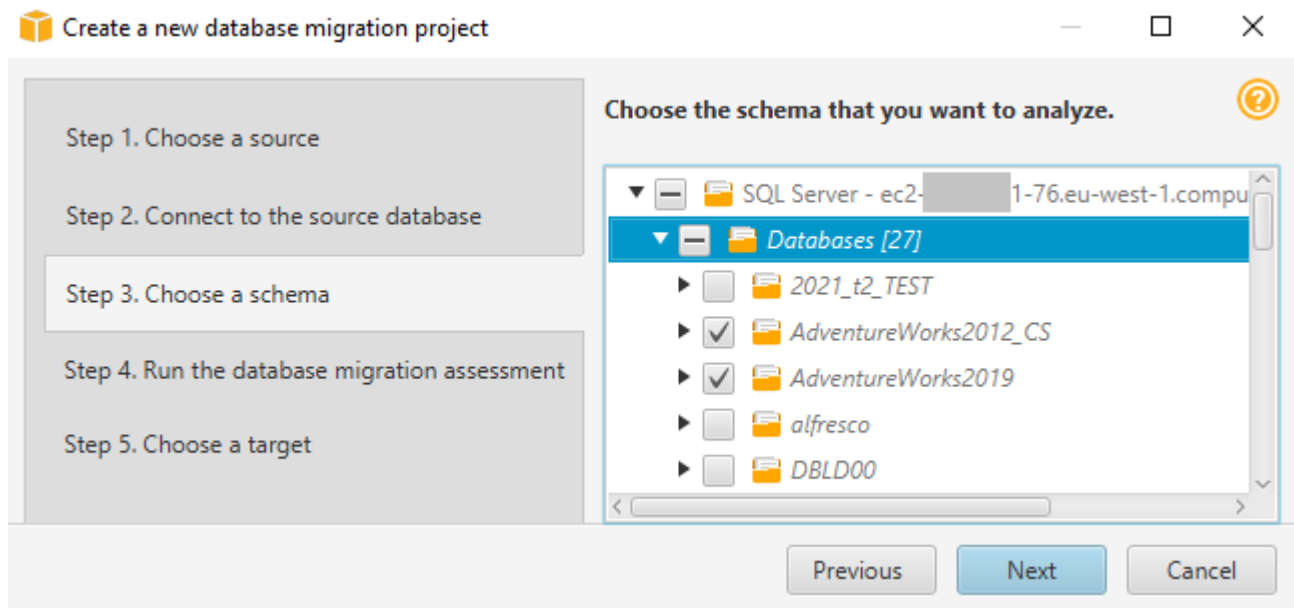
b. Wählen Sie Next (Weiter). Die Seite „Schema auswählen“ wird geöffnet.

3. Wählen Sie Ihr Datenbankschema.

- a. Aktivieren Sie das Kontrollkästchen für die Namen der Schemas, die Sie bewerten möchten, und wählen Sie dann das Schema selbst aus. Der Schemaname ist blau hervorgehoben, wenn er ausgewählt ist, und die Schaltfläche Weiter ist verfügbar.



- b. Wenn Sie mehrere Datenbankschemas überprüfen möchten, aktivieren Sie die Kontrollkästchen für alle Schemas und wählen Sie dann den übergeordneten Knoten aus. Für eine erfolgreiche Bewertung müssen Sie den übergeordneten Knoten auswählen. Wählen Sie beispielsweise für eine SQL Server-Quelldatenbank den Knoten Datenbanken aus. Der Name des übergeordneten Knotens ist blau hervorgehoben und die Schaltfläche Weiter ist verfügbar.



- c. Wählen Sie Weiter. AWS SCT analysiert Ihre Quelldatenbankschemas und erstellt einen Bewertungsbericht zur Datenbankmigration. Die Anzahl der Datenbankobjekte in Ihren Quelldatenbankschemas wirkt sich auf die Zeit aus, die für die Ausführung der Bewertung benötigt wird. Wenn der Vorgang abgeschlossen ist, wird die Seite „Bewertung der Datenbankmigration ausführen“ geöffnet.
4. Führen Sie die Bewertung der Datenbankmigration aus.
 - a. Sie können die Bewertungsberichte für verschiedene Migrationsziele überprüfen und vergleichen oder eine lokale Kopie der Bewertungsberichtsdateien für die weitere Analyse speichern.
 - b. Speichern Sie eine lokale Kopie des Bewertungsberichts zur Datenbankmigration. Wählen Sie Speichern, geben Sie dann den Pfad zu dem Ordner ein, in dem die Dateien gespeichert werden sollen, und wählen Sie Speichern. AWS SCT speichert die Bewertungsberichtsdateien im angegebenen Ordner.
 - c. Wählen Sie Next (Weiter). Die Seite „Wählen Sie ein Ziel“ wird geöffnet.
5. Wählen Sie Ihre Zieldatenbank aus.
 - a. Wählen Sie unter Ziel-Engine die Zieldatenbank-Engine aus, die Sie auf der Grundlage des Bewertungsberichts verwenden möchten.
 - b. Geben Sie Ihre Verbindungsinformationen für Ihre Zieldatenbank ein. Die Verbindungsparameter, die Sie sehen, hängen von der ausgewählten Zieldatenbank-Engine ab. Stellen Sie sicher, dass der für die Zieldatenbank angegebene Benutzer über die erforderlichen Berechtigungen verfügt. Weitere Informationen zu den erforderlichen

Berechtigungen finden Sie in den Abschnitten, in denen Berechtigungen für Zieldatenbanken in [Quellen für AWS SCT](#) und beschrieben [Berechtigungen für Amazon Redshift als Ziel](#) werden.

- c. Wählen Sie Fertig stellen. AWS SCT erstellt Ihr Projekt und fügt die Mapping-Regeln hinzu. Weitere Informationen finden Sie unter [Mapping-Regeln erstellen](#).

Jetzt können Sie das AWS SCT Projekt verwenden, um Ihre Quelldatenbankobjekte zu konvertieren.

AWS SCT Projekt speichern und öffnen

Gehen Sie wie folgt vor, um ein AWS Schema Conversion Tool Projekt zu speichern.

Speichern Ihres Projekts

1. Starten Sie den AWS Schema Conversion Tool.
2. Wählen Sie im Menü Datei die Option Projekt speichern.

AWS SCT speichert das Projekt in dem Ordner, den Sie bei der Erstellung des Projekts angegeben haben.

Gehen Sie wie folgt vor, um ein vorhandenes AWS Schema Conversion Tool Projekt zu öffnen.

Öffnen Ihres Projekts

1. Wählen Sie im Menü Datei die Option Projekt öffnen. Das Dialogfeld Öffnen wird angezeigt.
2. Wählen Sie den Projektordner und dann die Windows Script Component (*.sct) -Datei aus.
3. AWS SCT öffnet Ihr Projekt, stellt jedoch nicht automatisch eine Verbindung mit Ihrer Quell- und Ihrer Zieldatenbank her. Wählen Sie oben in Ihren Datenbankscheabäumen die Option Mit dem Server Connect aus, um eine Verbindung mit Ihrer Quell- und Ihrer Ziel-Datenbank herzustellen.

Wenn Sie ein Projekt öffnen, das in AWS SCT Version 1.0.655 oder früher gespeichert ist, werden automatisch Zuordnungsregeln für alle Quelldatenbankschemas zur Zieldatenbankplattform erstellt. Um weitere Zieldatenbankplattformen hinzuzufügen, löschen Sie vorhandene Zuordnungsregeln und erstellen Sie dann neue Zuordnungsregeln. Weitere Informationen zum Erstellen von Zuordnungsregeln finden Sie unter [Mapping-Regeln erstellen](#).

Hinzufügen von Datenbankservern zu einem AWS SCT Projekt

Sie können einem AWS Schema Conversion Tool Projekt mehrere Quell- und Zieldatenbankserver hinzufügen.

So fügen Sie Ihrem Projekt einen Server hinzu

1. Starten Sie den AWS Schema Conversion Tool.
2. Erstellen Sie ein neues Projekt oder öffnen Sie ein vorhandenes Projekt.
3. Wählen Sie Quelle hinzufügen aus dem Menü, um eine neue Quelldatenbank hinzuzufügen.
4. Wählen Sie eine Datenbankplattform und geben Sie die Anmeldeinformationen für die Datenbankverbindung an. Weitere Informationen zum Herstellen einer Verbindung mit einer Quelldatenbank finden Sie unter [Quellen für AWS SCT](#).

Verwenden Sie die folgende Vorgehensweise, um eine Verbindung mit Ihrer Datenbank herzustellen.

So stellen Sie eine Verbindung mit Ihrer Datenbank her

1. Öffnen Sie das Kontextmenü (rechte Maustaste) für einen Datenbankserver und wählen Sie dann Verbindung herstellen aus.

Sie können auch oben in Ihrer Datenbankschemastruktur die Option Mit dem Server Connect wählen.

2. Geben Sie das Passwort ein, um sich mit Ihrem Quelldatenbankserver zu verbinden.
3. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden kann.
4. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Gehen Sie wie folgt vor, um einen Datenbankserver aus Ihrem AWS SCT Projekt zu entfernen.

Um einen Datenbankserver zu entfernen

1. Wählen Sie den zu entfernenden Datenbankserver aus.
2. Öffnen Sie das Kontextmenü (rechte Maustaste) und wählen Sie dann Aus Projekt entfernen aus.

AWS SCT entfernt den ausgewählten Datenbankserver, alle Zuordnungsregeln, Konvertierungsergebnisse und andere Metadaten, die sich auf diesen Server beziehen.

AWS SCT im Offline-Modus ausführen

Sie können AWS Schema Conversion Tool im Offline-Modus laufen. Nachfolgend erfahren Sie, wie Sie mit einem vorhandenen AWS SCT Projekt arbeiten, wenn die Verbindung mit Ihrer Quelldatenbank getrennt ist.

AWS SCT benötigt keine Verbindung zu Ihrer Quelldatenbank, um die folgenden Operationen auszuführen:

- Fügen Sie Zuordnungsregeln hinzu.
- Erstellen Sie Bewertungsberichte zur Datenbankmigration.
- Konvertiert Datenbankschemas und Code.
- Bearbeiten Sie Ihren Quellcode und den konvertierten Code.
- Speichern Sie Ihren Quellcode und den konvertierten Code als SQL-Skripts in einer Textdatei.

Stellen Sie vor AWS SCT der Verwendung im Offline-Modus eine Verbindung zu Ihrer Quelldatenbank her, laden Sie Metadaten und speichern Sie Ihr Projekt. Öffnen Sie dieses Projekt oder trennen Sie die Verbindung zum Quelldatenbankserver, um es AWS SCT im Offline-Modus zu verwenden.

Um AWS SCT im Offline-Modus zu laufen

1. Starten Sie den AWS Schema Conversion Tool und erstellen Sie ein neues Projekt. Weitere Informationen finden Sie unter [Ein AWS SCT Projekt erstellen](#).
2. Fügen Sie einen Quelldatenbankserver hinzu und stellen Sie eine Verbindung zu Ihrer Quelldatenbank her. Weitere Informationen finden Sie unter [Hinzufügen von Datenbankservern zu einem AWS SCT Projekt](#).
3. Fügen Sie einen Zieldatenbankserver hinzu oder verwenden Sie eine virtuelle Zieldatenbankplattform. Weitere Informationen finden Sie unter [Verwendung virtueller Ziele](#).
4. Erstellen Sie eine Zuordnungsregel, um die Zieldatenbankplattform für Ihre Quelldatenbank zu definieren. Weitere Informationen finden Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#).
5. Wählen Sie Ansicht und dann Hauptansicht.

6. Wählen Sie im linken Bereich, in dem die Objekte Ihrer Quelldatenbank angezeigt werden, Ihre Quelldatenbankschemas aus. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie dann Schema laden aus. Dieser Vorgang lädt alle Quellschematadaten in IhrAWS SCT Projekt.

Mit den Vorgängen Bericht erstellen und Schema konvertieren werden auch alle Quellschematadaten in IhrAWS SCT Projekt geladen. Wenn Sie einen dieser Vorgänge über das Kontextmenü ausgeführt haben, überspringen Sie den Vorgang Schema laden.

7. Wählen Sie im Menü Datei die Option Projekt speichern, um die Metadaten der Quelldatenbank in Ihrem Projekt zu speichern.
8. Wählen Sie Verbindung zum Server trennen, um die Verbindung zu Ihrer Quelldatenbank zu trennen. Jetzt können Sie esAWS SCT im Offline-Modus verwenden.

AWS SCTBaumfilter verwenden

Zum Migrieren von Daten von einer Quelle in ein Ziel lädt AWS SCT alle Metadaten aus den Quell- und Zieldatenbanken in eine Baumstruktur. Diese Struktur wird in AWS SCT als Baumstrukturansicht im Hauptfenster angezeigt.

Einige Datenbanken können sehr viele Objekte in der Baumstruktur haben. Sie können Baumstrukturfilter in AWS SCT verwenden, um nach Objekten in den Quell- und Ziel-Baumstrukturen zu suchen. Wenn Sie einen Baumstrukturfilter verwenden, ändern Sie nicht die Objekte, die konvertiert werden, wenn Sie Ihre Datenbank konvertieren. Der Filter ändert nur das, was Sie im Baum sehen.

Baumstrukturfilter arbeiten mit Objekten, die AWS SCT vorab geladen hat. Mit anderen Worten: AWS SCT lädt bei der Suche keine Objekte aus der Datenbank. Dieser Ansatz bedeutet, dass die Baumstruktur im Allgemeinen weniger Objekte enthält, als in der Datenbank vorhanden sind.

Beachten Sie für Baumstrukturfilter Folgendes:

- Der Standardwert für den Filter ist ANY, was bedeutet, dass der Filter eine Namenssuche verwendet, um Objekte zu finden.
- Wenn Sie einen oder mehrere Objekttypen auswählen, sehen Sie nur diese Objekttypen in der Baumstruktur.
- Sie können den Filter verwenden, um unterschiedliche Symboltypen anzuzeigen, einschließlich Unicode, Leerzeichen und Sonderzeichen. "%" ist der Platzhalter für alle Zeichen.

- Nachdem Sie einen Filter angewendet haben, zeigt der Zähler nur die Anzahl der gefilterten Objekte.

So erstellen Sie einen Baumstrukturfilter

1. Öffnen Sie ein vorhandenes AWS SCT-Projekt.
2. Connect der Datenbank her, auf die Sie den Baumfilter anwenden möchten.
3. Wählen Sie das Filtersymbol.





Das Symbol für das Aufheben des Filters ist deaktiviert, da aktuell kein Filter angewendet wird.




4. Geben Sie die folgenden Informationen in das Dialogfeld Filter ein. Die Optionen im Dialogfeld unterscheiden je nach Datenbank-Engine.

AWS SCTFilter option	Action
Stufe	<p>Wählen Sie Kategorien, um Objekte nach Kategorien zu filtern.</p> <p>Wählen Sie Status, um Objekte nach Status zu filtern.</p>
Typ	<p>Wählen Sie für Kategorien in Level die Kategorien der gefilterten Objekte aus. Wählen Sie Beliebig geladen, um Objekte aus allen Kategorien anzuzeigen.</p> <p>Wählen Sie unter Status in Level den Status der gefilterten Objekte aus. Sie können eine der folgenden Optionen wählen:</p> <ul style="list-style-type: none"> • Konvertiert, um alle konvertierten Objekte anzuzeigen • Verfügt über Aktionen zum Anzeigen aller Objekte, bei denen Konvertierungsprobleme auftreten • Verschlüsselt, um alle verschlüsselten Objekte anzuzeigen
Bedingung	Wählen Sie für Kategorien in Level die Filterbedingung zwischen Gefällt mir und Gefällt mir nicht.

AWS SCTFilter option	Action
	Für Status in Level ist die Filterbedingungsoption nicht verfügbar.
Wert	<p>Geben Sie für Kategorien unter Level den Wert ein, um den Baum nach diesem Wert zu filtern.</p> <p>Verwenden Sie die Prozentzahl (%) als Platzhalter, um alle Objekte anzuzeigen.</p> <p>Wählen Sie für Status in Level den Wert zwischen Wahr und Falsch.</p>
Und/oder	Wählen SieAND unsereOR logischen Operatoren, um mehrere Filterklauseln anzuwenden.

 Filter
×

Specify multiple filters or filter values for schemas or any other objects. Use % as a wildcard. 

		Level	Type	Condition	Value	And/Or	
+		<input type="checkbox"/>	Categories	Any loaded	Like	%dbo%	AND
+		<input type="checkbox"/>	Categories	Tables	Like	%tmp%	AND
+		<input type="checkbox"/>	Statuses	Mapped	Value	True	

Any loaded like %dbo% AND Tables like %tmp% AND mapped value true

- Wählen Sie Neue Klausel hinzufügen, um eine zusätzliche Filterklausel hinzuzufügen. AWS SCTkann mehrere Filterklauseln mithilfeAND unsererOR logischen Operatoren anwenden.
- Wählen Sie Apply (Anwenden) aus. Nachdem Sie Apply ausgewählt haben, wird das Symbol für die Aufhebung des Filters (neben dem Filtersymbol) aktiviert. Verwenden Sie dieses Symbol, wenn Sie die angewendeten Filter entfernen möchten.

7. Wählen Sie Close (Schließen), um das Dialogfeld zu schließen.

Wenn Sie das Schema filtern, das in der Struktur angezeigt wird, ändern Sie nicht die Objekte, die konvertiert werden, wenn Sie das Schema konvertieren. Der Filter ändert nur das, was in der Struktur zu sehen ist.

Import einer Dateiliste für den Baumfilter

Sie können eine Datei mit kommagetrennten Werten (CSV) mit Semikolontrennzeichen oder eine JSON-Datei importieren, die Namen oder Werte enthält, die der Baumfilter verwenden soll. Öffnen Sie ein vorhandenes AWS SCT Projekt, stellen Sie eine Verbindung mit der Datenbank her, auf die Sie den Baumfilter anwenden möchten, und wählen Sie dann das Filtersymbol.

Um ein Beispiel der Datei herunterzuladen, wählen Sie Vorlage herunterladen. Geben Sie den Dateinamen ein und wählen Sie Speichern.

Um Ihre vorhandenen Filtereinstellungen herunterzuladen, wählen Sie Exportieren. Geben Sie den Dateinamen ein und wählen Sie Speichern.

Um eine Dateiliste für den Baumfilter zu importieren, wählen Sie Importieren. Wählen Sie eine zu importierende Datei aus und wählen Sie dann Open. Wählen Sie Apply und dann Close.

CSV-Dateien verwenden Semikolon als Trennzeichen und haben das folgende Format:

- `object_type` ist der Objekttyp, nach dem Sie suchen möchten.
- `database_name` ist der Name der Datenbank, in der das Objekt vorhanden ist.
- `schema_name` ist der Name des Schemas, in dem das Objekt vorhanden ist.
- `object_name` ist der Name des Objekts.
- `import_type` gibt an `include` oder `exclude` dieses Element aus dem Filter.

Verwenden Sie JSON-Dateien, um komplexe Filterfälle zu beschreiben, z. B. verschachtelte Regeln. JSON-Dateien haben folgendes Format:

- `filterGroupType` ist der Typ der Filterregel (AND oder OR logischen Operatoren), der für mehrere Filterklauseln gilt.
- `filterCategory` ist die Ebene des Filters (Kategorien oder Status).
- `names` ist die Liste der Objektnamen, die für den Filter Kategorien gilt.

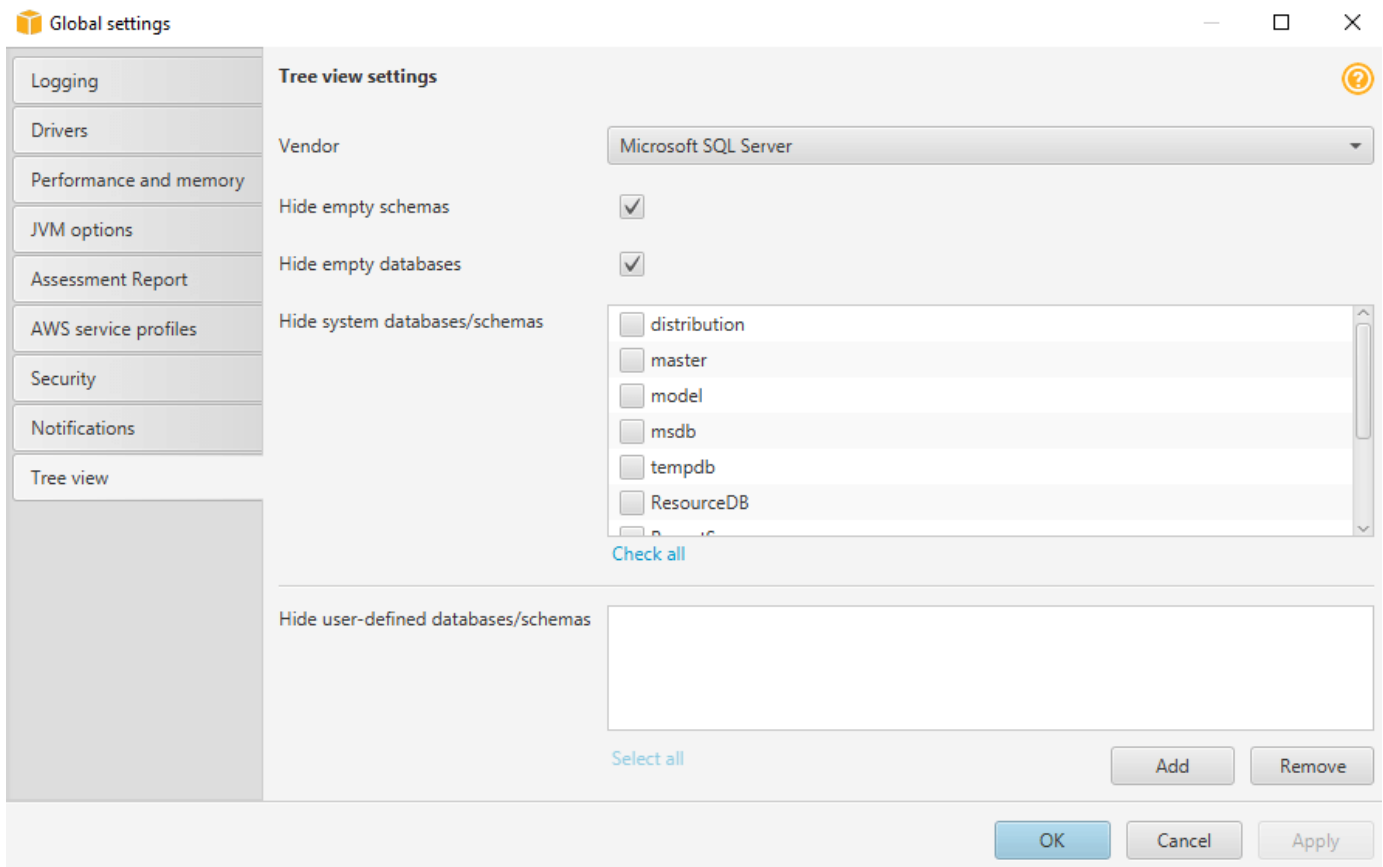
- `filterCondition` ist die Filterbedingung (LIKE oder NOT LIKE), die für den Filter Kategorien gilt.
- `transformName` ist der Statusname, der für den Statusfilter gilt.
- `value` ist der Wert, nach dem der Baum gefiltert werden soll.
- `transformValue` ist der Wert des Filters (TRUE oder FALSE), der für den Statusfilter gilt.

Schemas in der AWS SCT Baumansicht ausblenden

Mit den Einstellungen der Baumstrukturansichten geben Sie an, welche Schemata und Datenbanken in der AWS SCT-Baumstrukturansicht angezeigt werden sollen. Sie können leere Schemas, leere Datenbanken, System-Datenbanken und benutzerdefinierten Datenbanken und Schemas ausblenden.

So blenden Sie Datenbanken und Schemas in der Baumstrukturansicht aus

1. Öffnen Sie ein AWS SCT-Projekt.
2. Stellen Sie eine Verbindung mit dem Datenspeicher her, den Sie in der Baumstrukturansicht anzeigen möchten.
3. Wählen Sie Einstellungen, Globale Einstellungen, Baumansicht.



4. Führen Sie im Abschnitt Einstellungen für die Strukturansicht folgende Schritte aus:

- Wählen Sie als Anbieter die Datenbankplattform aus.
- Wählen Sie Leere Schemas ausblenden, um leere Schemas für die ausgewählte Datenbankplattform auszublenden.
- Wählen Sie Leere Datenbanken ausblenden, um leere Datenbanken für die ausgewählte Datenbankplattform auszublenden.
- Wählen Sie unter Systemdatenbanken/Schemas ausblenden die Option Systemdatenbanken und Schemas nach Namen aus, um sie auszublenden.
- Geben Sie unter Benutzerdefinierte Datenbanken/Schemas ausblenden die Namen der benutzerdefinierten Datenbanken und Schemas ein, die Sie ausblenden möchten, und wählen Sie dann Hinzufügen. Bei den Namen wird nicht zwischen Groß- und Kleinschreibung unterschieden.

5. Wählen Sie OK.

Erstellung und Überprüfung des Bewertungsberichts zur Datenbankmigration

Der Bewertungsbericht zur Datenbankmigration fasst alle Aktionspunkte für Schemas zusammen, die nicht automatisch in die Engine Ihrer Amazon RDS-DB-Zielinstanz konvertiert werden können. Im Bericht ist zudem eine Aufwandsschätzung für das Schreiben des äquivalenten Codes für Ihre Ziel-DB-Instance enthalten.

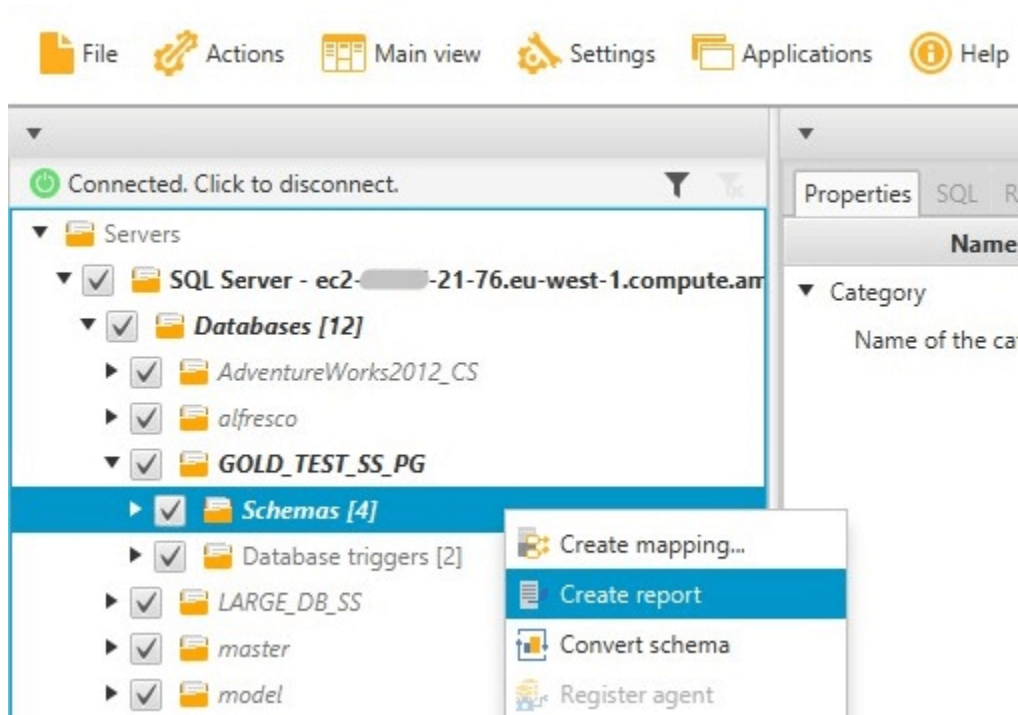
Sie können einen Bewertungsbericht zur Datenbankmigration erstellen, nachdem Sie Ihrem Projekt die Quelldatenbanken und Zielplattformen hinzugefügt und die Zuordnungsregeln festgelegt haben.

So erstellen Sie den Bewertungsbericht zur Datenbankmigration und zeigen diesen an

1. Stellen Sie sicher, dass Sie eine Zuordnungsregel für das Quelldatenbankschema erstellt haben, für das Sie einen Bewertungsbericht erstellen möchten. Weitere Informationen finden Sie unter [Neue Zuordnungsregel hinzufügen](#).
2. Wählen Sie im Menü Ansicht die Option Hauptansicht.
3. Wählen Sie im linken Bereich, in dem das Schema aus Ihrer Quelldatenbank angezeigt wird, Schemaobjekte aus, für die Sie einen Bewertungsbericht erstellen möchten.

Stellen Sie sicher, dass Sie die Kontrollkästchen für alle Schemaobjekte aktiviert haben, für die Sie einen Bewertungsbericht erstellen möchten.

4. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie dann Bericht erstellen aus.



Die Bewertungsberichtansicht wird geöffnet.

5. Wählen Sie den Tab Aktionselemente.

Auf der Registerkarte Aktionselemente wird eine Liste von Elementen angezeigt, die das Schema beschreiben und nicht automatisch konvertiert werden können. Wählen Sie eines der Aktionspunkte in der Liste aus. AWS SCT hebt das Element aus Ihrem Schema hervor, für das sich das Aktionselement bezieht, wie im Folgenden gezeigt.

The screenshot displays the AWS Schema Conversion Tool interface. The left pane shows a tree view of the source database structure, including servers, databases, tables, and procedures. The main pane is divided into two sections: a list of conversion issues and a detailed view of a specific issue.

Issues List:

- Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required. Recommended action: Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table.
- Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically.
- Issue 794:** MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype.
- Issue 826:** Check the default value for a DateTime variable.
- Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision.
- Issue 9997:** Unable to resolve objects.
- Issue 690:** MySQL doesn't support table types.
- Issue 811:** Unable to convert functions.

SQL Procedure Definition:

```

1 create procedure POSITION_UPDATE_CASH_CGT_BULK
2   @InputPosNo tinyint readonly
3   , @posFlags bigint = 0
4   , @posFlagsMask bigint = 0
5 AS
6 update p
7 set   p.Flags = p.Flags & (~ @posFlagsMask ) | @posFlags
8 from   Position p
9       inner join @InputPosNo ipn on p.PosNo = ipn.F_POSNO
10
11 return 0

```

6. Wählen Sie die Registerkarte Summary aus.

Auf der Registerkarte Summary wird eine Zusammenfassung des Berichts zur Bewertung der Datenbankmigration angezeigt. Es wird sowohl die Anzahl der automatisch konvertierten Elemente angezeigt als auch die Anzahl der Elemente, die nicht automatisch konvertiert werden konnten. Die Übersicht enthält auch eine Schätzung der Zeit, die es dauert, Schemata in Ihrer Ziel-DB-Instance zu erstellen, die denen in Ihrer Quelldatenbank gleichen.

Der Abschnitt Lizenzbewertung und Cloud-Support enthält Informationen zum Verschieben Ihres vorhandenen lokalen Datenbankschemas in eine Amazon RDS-DB-Instance, auf der dieselbe Engine ausgeführt wird. Wenn Sie beispielsweise die Lizenztypen ändern möchten, erfahren Sie in diesem Abschnitt des Berichts, welche Funktionen aus Ihrer aktuellen Datenbank entfernt werden müssen.

Ein Beispiel für eine Zusammenfassung eines Bewertungsberichts wird im Folgenden gezeigt.

Summary | Action Items

Save to CSV | Save to PDF

Database migration assessment report

Source database: GOLD_TEST_SS_PG:21-76.eu-west-1.compute.amazonaws.com/GOLD_TEST_SS_PG:1433
 Microsoft SQL Server 2019 (RTM-CU10) (KB5001090) - 15.0.4123.1 (X64) Mar 22 2021 18:10:24
 Copyright (C) 2019 Microsoft Corporation
 Enterprise Edition: Core-based Licensing (64-bit) on Windows Server 2019 Datacenter 10.0 <X64> (Build 17763:) (Hypervisor)
 Case sensitivity: Off

Executive summary

We completed the analysis of your Microsoft SQL Server source database and estimate that 90% of the database storage objects and 77% of database code objects can be converted automatically or with minimal changes if you select Amazon RDS for PostgreSQL as your migration target. Database storage objects include schemas, tables, table constraints, indexes, types, table types, sequences, synonyms and xml schema collections. Database code objects include triggers, views, procedures, scalar functions, inline functions, table-valued functions and database triggers. Based on the source code syntax analysis, we estimate 94% (based on # lines of code) of your code can be converted to Amazon RDS for PostgreSQL automatically. To complete the migration, we recommend 3,300 conversion action(s) ranging from simple tasks to medium-complexity actions to complex conversion actions.

Migration guidance for database objects that could not be converted automatically can be found [here](#)

Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

Figure: Conversion statistics for database storage objects

Object Type	Count	Automatically converted	With simple actions	With medium-complexity actions	With complex actions
Schema (4: 4/0/0/0)	4	100%	0%	0%	0%
Table (323: 276/8/2/37)	323	85%	2%	11%	2%
Constraint (157: 152/2/0/3)	157	97%	2%	0%	0%
Index (63: 36/22/0/5)	63	57%	35%	8%	0%
Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Sequence (14: 7/7/0/0)	14	50%	50%	0%	0%
Synonym (5: 0/0/0/5)	5	0%	0%	100%	0%
Table Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Xml schema collection (5: 1/0/0/4)	5	20%	80%	0%	0%

7. Wählen Sie die Registerkarte Summary und anschließend Save to PDF aus. Der Bewertungsbericht zur Datenbankmigration wird als PDF-Datei gespeichert. Die PDF-Datei enthält sowohl die Zusammenfassung als auch Informationen zu den Aktionselementen.

Sie können auch „In CSV speichern“ wählen, um den Bericht als CSV-Datei zu speichern. Wenn Sie diese Option wählen, werden drei CSV-Dateien AWS SCT erstellt. Diese Dateien enthalten die folgende Informationen:

- Eine Liste der Elemente für Konversionsaktionen mit empfohlenen Aktionen.
- Eine Zusammenfassung der Konvertierungsaktionselemente mit einer Schätzung des Aufwands, der erforderlich ist, um ein Vorkommen des Aktionselements zu konvertieren.
- Eine Zusammenfassung mit einer Reihe von Aktionspunkten, sortiert nach der geschätzten Zeit bis zur Konvertierung.

Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

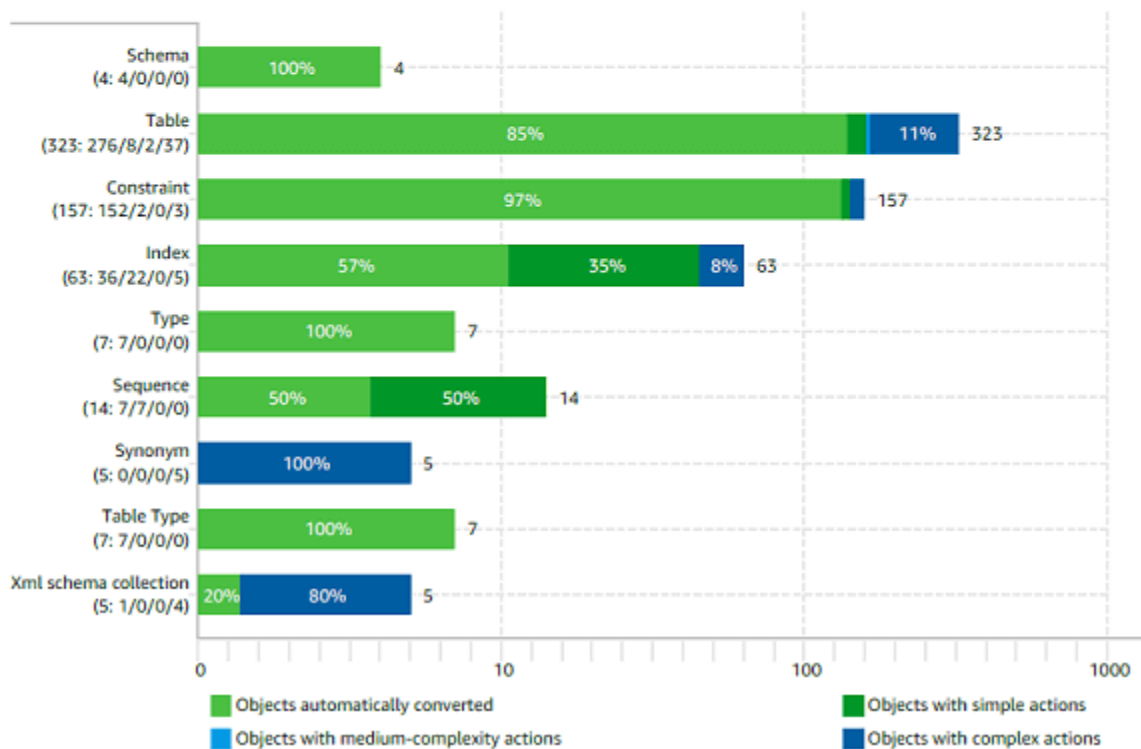
We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

Figure: Conversion statistics for database storage objects

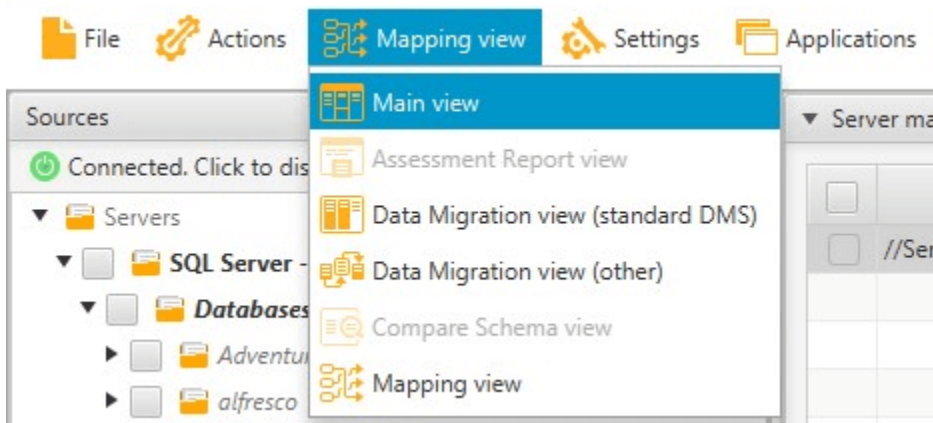


Konvertieren Ihres Schemas

Nachdem Sie Ihrem Projekt Quell- und Zieldatenbanken hinzugefügt und Zuordnungsregeln erstellt haben, können Sie Ihre Quelldatenbankschemas konvertieren. Führen Sie die folgenden Schritte durch, um ein Schema zu konvertieren.

Konvertieren Ihres Schemas

1. Wählen Sie Ansicht und dann Hauptansicht.



2. Aktivieren Sie im linken Bereich, in dem das Schema aus Ihrer Quelldatenbank angezeigt wird, das Kontrollkästchen für den Namen des zu konvertierenden Objekts. Klicken Sie als Nächstes auf dieses Objekt. AWS SCT markiert den Objektnamen blau. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie Schema konvertieren aus.

Um mehrere Datenbankobjekte zu konvertieren, aktivieren Sie die Kontrollkästchen für alle Objekte. Wählen Sie als Nächstes den übergeordneten Knoten aus. Für Tabellen ist der übergeordnete Knoten beispielsweise Tabellen. Stellen Sie sicher, dass der Name des übergeordneten Knotens blau hervorgehoben ist. Öffnen Sie das Kontextmenü (rechte Maustaste) für den übergeordneten Knoten und wählen Sie Schema konvertieren aus.

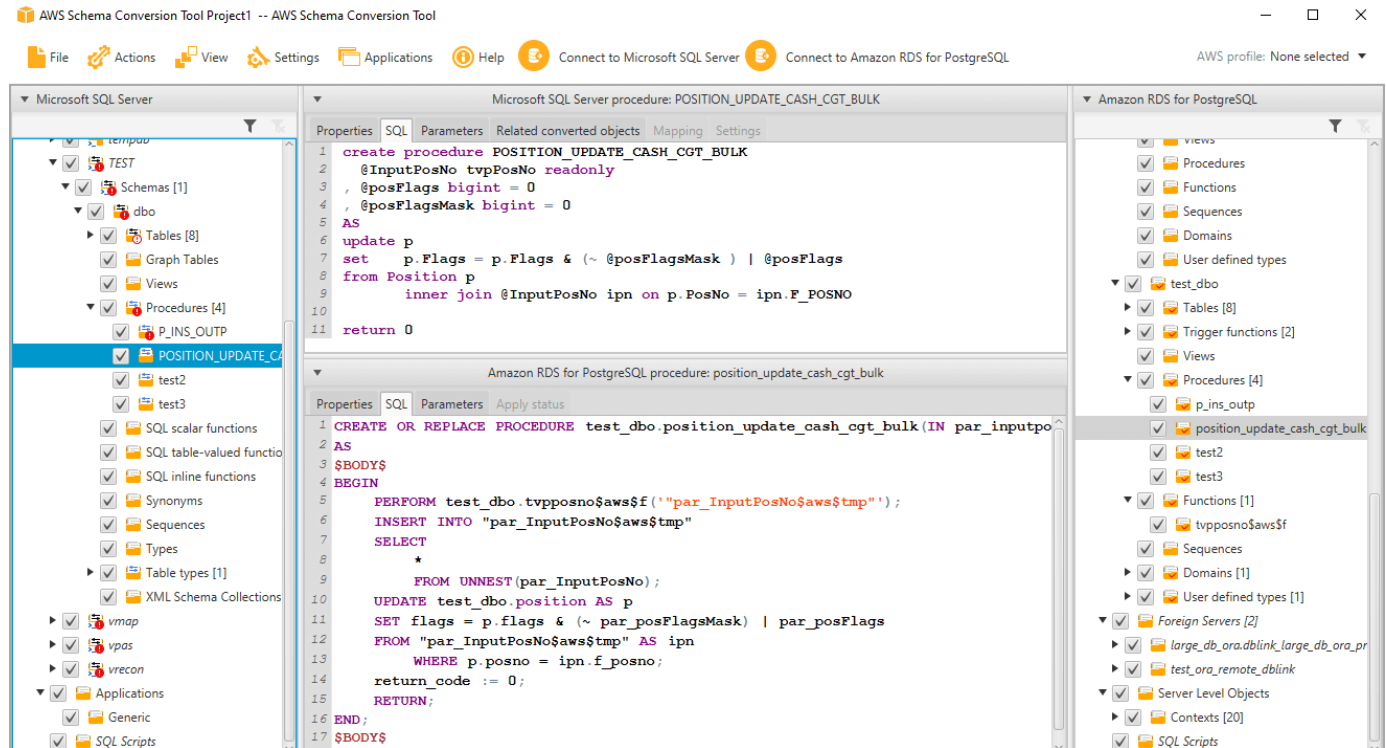
The screenshot displays the AWS Schema Conversion Tool interface. At the top, there is a menu bar with icons for File, Actions, Main view, Settings, Applications, Help, Add source, and Add target. The main workspace is divided into three panes:

- Left Pane:** A tree view showing a connection to an SQL Server instance. Under 'Servers', the 'TEST' database is selected. A context menu is open over 'TEST', with 'Convert schema' highlighted.
- Top Right Pane:** Shows the 'Properties' tab for the selected 'TEST' database. It lists details such as 'Created' (2021-09-06 09:56:08.26), 'Object name' (Name: TEST, compatibility-level: 100, collation-name: SQL_Latin1_General_CP1_CI_AS).
- Bottom Right Pane:** Shows the 'Properties' tab for the converted schema, displaying 'Category' (Name: <Aurora_MySQL (virtual)>).

- Nachdem das AWS SCT das Schema konvertiert hat, können Sie das vorgeschlagene Schema im Bereich rechts neben Ihrem Projekt anzeigen.

Zu diesem Zeitpunkt wird kein Schema auf Ihre Zieldatenbankinstanz angewendet. Das geplante Schema ist Teil Ihres Projekts. Wenn Sie ein konvertiertes Schemaelement auswählen, können Sie den Befehl für das geplante Schema im Bereich unten in der Mitte für Ihre Zieldatenbankinstanz sehen.

Sie können das Schema in diesem Fenster bearbeiten. Das bearbeitete Schema wird als Teil Ihres Projekts gespeichert und in die Zieldatenbankinstanz geschrieben, wenn Sie Ihr konvertiertes Schema anwenden möchten.



Anwenden des konvertierten Schemas auf Ihre Ziel-DB-Instanz

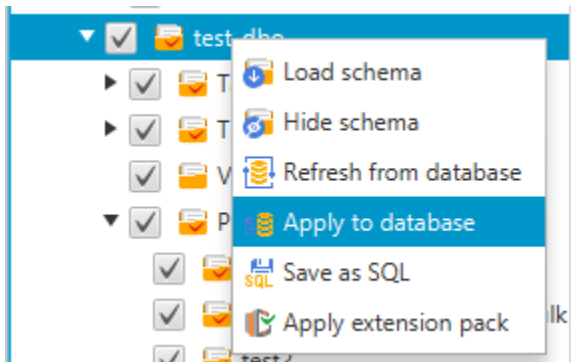
Sie können das konvertierte Datenbankschema auf Ihre DB-Ziel-Instanz anwenden. Nachdem das Schema auf Ihre Ziel-DB-Instanz angewendet wurde, können Sie das es basierend auf den Aktionselementen im Bewertungsbericht zur Datenbankmigration aktualisieren.

⚠ Warning

Das folgende Verfahren überschreibt das vorhandene Zielschema. Achten Sie darauf, Schemas nicht unbeabsichtigt zu überschreiben. Achten Sie darauf, dass Sie Schemas in Ihrer Ziel-DB-Instanz, die Sie bereits geändert haben, nicht überschreiben, oder Sie überschreiben diese Änderungen.

Um das konvertierte Datenbankschema auf Ihre Zieldatenbankinstanz anzuwenden

1. Wählen Sie oben im rechten Bereich Ihres Projekts die Option Mit dem Server Connect, um eine Verbindung zu Ihrer Zieldatenbank herzustellen. Wenn Sie mit Ihrer Zieldatenbank verbunden sind, überspringen Sie diesen Schritt.
2. Wählen Sie das Schema-Element im rechten Bereich Ihres Projekt aus, der das geplante Schema für Ihre Ziel-DB-Instance anzeigt.
3. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Schema-Element und wählen Sie Apply to database aus.



Das konvertierte Schema wird auf Ihre Ziel-DB-Instance angewendet.

Speichern von AWS Serviceprofilen in AWS SCT

Sie können Ihre AWS Anmeldeinformationen in AWS SCT speichern. AWS SCT verwendet Ihre Anmeldeinformationen, wenn Sie Funktionen verwenden, die AWS Dienste integriert sind. AWS SCT lässt sich beispielsweise in Amazon S3, AWS Lambda, Amazon Relational Database Service (Amazon RDS) und AWS Database Migration Service (AWS DMS) integrieren.

AWS SCT fragt Sie nach Ihren AWS Anmeldeinformationen, wenn Sie auf eine Funktion zugreifen, für die diese erforderlich sind. Sie können die Anmeldeinformationen in den globalen Anwendungseinstellungen speichern. Sie können dann die gespeicherten Anmeldeinformationen auswählen, wenn Sie von AWS SCT zur Eingabe aufgefordert werden.

In den globalen Anwendungseinstellungen können Sie verschiedene Gruppen von AWS Anmeldeinformationen speichern. Sie können beispielsweise einen Anmeldeinformationssatz, den Sie in Testszenarien nutzen, und einen weiteren Anmeldeinformationssatz, den Sie in Produktionsszenarien verwenden, speichern. Sie können auch verschiedene Anmeldeinformationen für verschiedene AWS-Regionen speichern.

AWSAnmeldeinformationen speichern

Gehen Sie wie folgt vor, umAWS Anmeldeinformationen global zu speichern.

UmAWS Anmeldeinformationen zu speichern

1. Starten Sie den AWS Schema Conversion Tool.
2. Öffnen Sie das Menü Einstellungen und wählen Sie dann Globale Einstellungen. Das Dialogfeld Global settings wird angezeigt.
3. Wählen Sie AWSDienstprofile und dann NeuesAWS Dienstprofil hinzufügen aus.
4. Geben Sie IhreAWS Daten wie folgt ein.

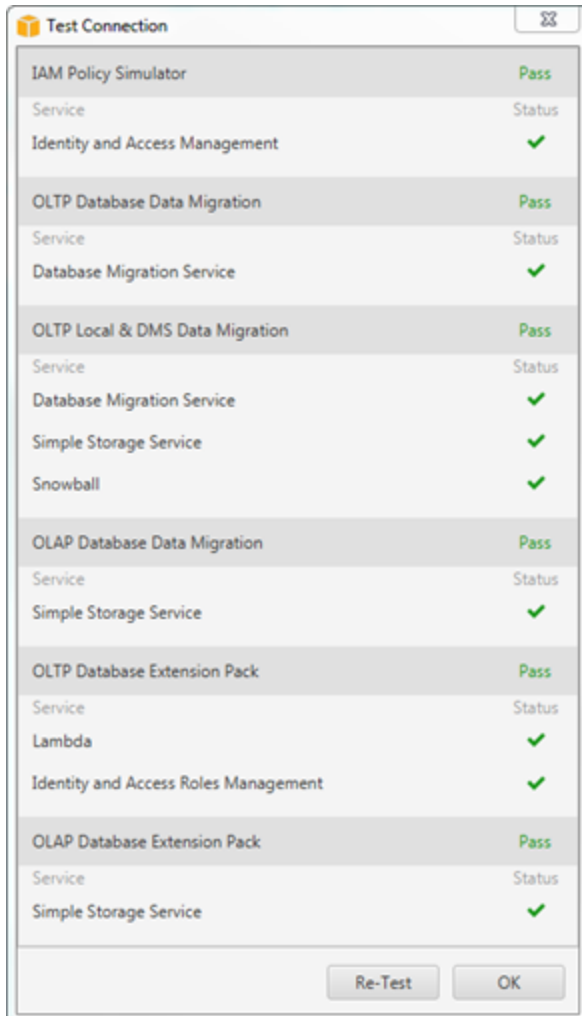
AWS SCT-Option	Action
Profilname	Geben Sie einen Namen für Ihr Profil an.
AWSzugriffsschlüssel	Geben Sie IhrenAWS Zugangsschlüssel ein.
AWSgeheimer Schlüssel	Geben Sie IhrenAWS geheimen Zugriffsschlüssel ein. Weitere Informationen zuAWS Zugriffsschlüsseln finden Sie unter Verwalten von Zugriffsschlüsseln im IAM-Benutzerhandbuch.
Region	Wählen Sie dasAWS-Region für Ihr Profil.
Amazon S3-Bucket-Ordner	Wählen Sie den Amazon S3-Bucket für Ihr Profil aus. Sie müssen einen Bucket nur angeben, wenn Sie eine Funktion verwenden, die eine Verbindung zu Amazon S3 herstellt. Weitere Informationen zu den erforderlichen Berechtigungen finden Sie unter Berechtigungen für die Verwendung desAWS Dienstprofils .

Wählen Sie Use FIPS endpoint for S3, wenn Sie die Sicherheitsanforderungen für die Federal Information Processing Standard (FIPS) erfüllen müssen. FIPS-Endpunkte sind in den folgendenAWS -Regionen verfügbar:

- Region USA Ost (Nord-Virginia)
- Region USA Ost (Ohio)

- Region US West (N. California)
 - Region USA West (Oregon)
5. Wählen Sie Verbindung testen, um zu überprüfen, ob Ihre Anmeldeinformationen korrekt und aktiv sind.

Das Dialogfeld Verbindung testen wird angezeigt. Für jeden Service, der mit Ihrem Profil verknüpft ist, wird der Status angegeben. Die Option Pass gibt an, dass das Profil erfolgreich auf den Service zugreifen kann.



6. Nachdem Sie Ihr Profil konfiguriert haben, wählen Sie Save, um Ihr Profil zu speichern, oder Cancel, um die Änderungen zu verwerfen.
7. Wählen Sie OK, um das Dialogfeld „Globale Einstellungen“ zu schließen.

Festlegen des Standardprofils für ein Projekt

Sie können ein Standardprofil für ein AWS SCT-Projekt vorgeben. Dadurch werden die im Profil gespeicherten AWS Anmeldeinformationen mit dem Projekt verknüpft. Öffnen Sie das Projekt und führen Sie die folgenden Schritte aus, um das Standardprofil festzulegen.

So legen Sie das Standardprofil für ein Projekt fest

1. Starten Sie den AWS Schema Conversion Tool und erstellen Sie ein neues Projekt.
2. Wählen Sie im Menü Einstellungen die Option Projekteinstellungen aus. Das Dialogfeld Projekteinstellungen wird angezeigt.
3. Wählen Sie die Registerkarte Projektumgebung.
4. Wählen Sie Neues AWS Dienstprofil hinzufügen, um ein neues Profil hinzuzufügen. Wählen Sie dann unter AWS Serviceprofil das Profil aus, das Sie mit dem Projekt verknüpfen möchten.
5. Wählen Sie OK, um das Dialogfeld mit den Projekteinstellungen zu schließen. Zum Verwerfen Ihrer Änderungen können Sie auf Cancel klicken.

Berechtigungen für die Verwendung des AWS Dienstprofils

Die folgenden Berechtigungen sind erforderlich, um von Ihrem AWS Serviceprofil aus auf Ihren Amazon S3-Bucket zuzugreifen:

- `s3:PutObject`— zum Hinzufügen von Objekten in Ihrem Amazon S3-Bucket.
- `s3:DeleteObject`— um die Null-Version eines Objekts zu entfernen und eine Löschmarkierung einzufügen, die zur aktuellen Version des Objekts wird.
- `s3:ListBucket`— um bis zu 1.000 Objekte aus Ihrem Amazon S3-Bucket zurückzugeben.
- `s3:GetObject`— zum Abrufen von Objekten aus Ihrem Amazon S3-Bucket.

Das folgende Codebeispiel veranschaulicht, wie Sie Ihrem Benutzer diese Berechtigungen gewähren.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
    ],
    "Resource": [
        "*"
    ]
}
]
```

Verwenden von AWS Secrets Manager

AWS SCT kann Datenbankverbindungsdaten verwenden, die in AWS Secrets Manager gespeichert sind. Sie können alle Werte im Dialogfeld für die Datenbankverbindung von Secrets Manager aus eingeben. Um Secrets Manager zu verwenden, stellen Sie sicher, dass Sie AWS Profile in der Speicher AWS Schema Conversion Tool.

Weitere Informationen zur Verwendung AWS Secrets Manager finden Sie unter [Was ist AWS Secrets Manager?](#) im AWS Secrets Manager Benutzerhandbuch. Weitere Informationen zum Speichern von AWS Profilen finden Sie unter [Speichern von AWS Serviceprofilen in AWS SCT](#).

So rufen Sie Datenbankverbindungsdaten von Secrets Manager ab

1. Starten Sie den AWS Schema Conversion Tool und erstellen Sie ein neues Projekt.
2. Wählen Sie Quelle hinzufügen oder Ziel hinzufügen, um Ihrem Projekt eine neue Datenbank hinzuzufügen.
3. Wählen Sie eine Datenbankplattform und klicken Sie dann auf Weiter.
4. Wählen Sie für AWS Secret das Secret aus, das Sie verwenden möchten.
5. Wählen Sie „Auffüllen“. Füllen AWS SCT Sie dann alle Werte im Dialogfeld für die Datenbankverbindung aus.
6. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Datenbank hergestellt werden kann.
7. Wählen Sie Connect, um eine Verbindung zu Ihrer Datenbank herzustellen.

AWS SCT unterstützt Secrets mit der folgenden Struktur.

```
{
  "username": "secret_user",
  "password": "secret_password",
  "engine": "oracle",
  "host": "secret_host.eu-west-1.compute.amazonaws.com",
  "port": "1521",
  "dbname": "ora_db"
}
```

In dieser Struktur sind die `password` Werte `username` und erforderlich, und alle anderen Werte sind optional. Stellen Sie sicher, dass die Werte, die Sie in Secrets Manager speichern, alle Datenbankmeldeinformationen enthalten.

Speichern von Datenbankpasswörtern

Sie können ein Datenbankpasswort oder SSL-Zertifikat im AWS SCT-Cache speichern. Um ein Passwort zu speichern, wählen Sie `Store Password`, (`Passwort speichern`) aus, wenn Sie eine Verbindung erstellen.

Das Passwort wird unter Verwendung des zufällig generierten Token in der Datei `seed.dat` verschlüsselt. Das Passwort wird dann mit dem Benutzernamen in der Cache-Datei gespeichert. Ist die Datei `seed.dat` nicht mehr vorhanden oder beschädigt, ist das Datenbankpasswort möglicherweise nicht richtig verschlüsselt. In diesem Fall kann keine Verbindung hergestellt werden.

Verwenden der UNION ALL-Ansicht für Projekte mit partitionierten Tabellen

Wenn eine Quelltable partitioniert ist, werden n Zieltabellen AWS SCT erstellt, wobei n die Anzahl der Partitionen in der Quelltable ist. AWS SCT erstellt eine UNION ALL-Ansicht über den Zieltabellen, um die Quelltable darzustellen. Wenn Sie einen AWS SCT Datenextraktor verwenden, um Ihre Daten zu migrieren, werden die Quelltablepartitionen durch separate Unteraufgaben parallel extrahiert und geladen.

So verwenden Sie Union Alle View für ein Projekt

1. Starten AWS SCT. Erstellen Sie ein neues Projekt oder öffnen Sie ein vorhandenes AWS SCT Projekt.
2. Wählen Sie im Menü `Einstellungen` die Option `Konvertierungseinstellungen` aus.

3. Wählen Sie ein Paar von OLAP-Datenbanken aus der Liste oben aus.
4. Gesamtansicht vereinigen?

Conversion settings

Choose vendor pair: Microsoft SQL Server - Amazon Redshift

Only mapped pairs

Conversion settings

Conversion optimization

Optimization strategies

Transformation rules

Settings that take effect during conversion

How detailed should comments be in the converted SQL? Errors only

Maximum number of tables for target Amazon Redshift cluster: Auto

Use Union all view? 368

Use compression encoding

Use compression encoding for KEY fields

OK Cancel Apply

5. Wählen Sie OK, um die Einstellungen zu speichern und das Dialogfeld „Konvertierungseinstellungen“ zu schließen.

Tastenkombinationen für AWS SCT

Im Folgenden sind die Tastenkombinationen aufgeführt, die Sie verwenden können AWS SCT.

Tastaturbefehl	Beschreibung
Strg+N	Erstellt ein neues Projekt.
Strg+O	Öffnet ein vorhandenes Projekt.
Strg+S	Speichert ein geöffnetes Projekt.
Strg+W	Erstellt ein neues Projekt mithilfe des Assistenten.
Strg+M	Erstellen Sie ein neues Multiserver-Assessment.
Strg+L	Fügen Sie eine neue Quelldatenbank hinzu.

Tastaturbefehl	Beschreibung
Strg+R	Fügen Sie eine neue Zieldatenbank hinzu.
Strg+F4	Schließt ein geöffnetes Projekt.
F1	Öffnen Sie das AWS SCTBenutzerhandbuch.

Erste Schritte mit AWS SCT

Sie können das AWS Schema Conversion Tool (AWS SCT), um das Schema für eine Quelldatenbank zu konvertieren. Die Quelldatenbank kann eine selbstverwaltete Engine sein, die lokal oder auf einer Amazon EC2 EC2-Instance ausgeführt wird. Sie können Ihr Quellschema in ein Schema für jede unterstützte Datenbank konvertieren, die von AWS. Die AWS SCT-Anwendung stellt eine projektbasierte Benutzeroberfläche bereit.

Nahezu alle Arbeiten, die Sie mit AWS SCT durchführen, beginnen mit den folgenden Schritten:

1. Installieren AWS SCT. Weitere Informationen finden Sie unter [Installation, Überprüfung und Aktualisierung AWS SCT](#).
2. Installieren einer AWS SCT Agent, falls erforderlich. AWS SCT Agenten sind nur für bestimmte Migrationsszenarien erforderlich, z. B. zwischen heterogenen Quellen und Zielen. Weitere Informationen finden Sie unter [Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift](#).
3. Machen Sie sich mit der Benutzeroberfläche von vertraut AWS SCT. Weitere Informationen finden Sie unter [Verwendung der AWS SCT Benutzeroberfläche](#).
4. Erstellen Sie ein AWS SCT-Projekt. Stellen Sie eine Verbindung mit Ihren Quell- und Zieldatenbanken her. Weitere Information zur Herstellung einer Verbindung zu Ihrer Quelldatenbank finden Sie unter [Quellen für AWS SCT](#).
5. Erstellen Sie Zuordnungsregeln. Weitere Informationen zu Zuweisungs-Regeln finden Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#).
6. Führen Sie den Database Migration Assessment Report (Bewertungsbericht zur Datenbankmigration) aus und zeigen Sie ihn an. Weitere Informationen zum Bewertungsbericht finden Sie unter [Erstellung und Überprüfung des Bewertungsberichts zur Datenbankmigration](#).
7. Wandeln Sie die Quelldatenbank Schemas um. Es gibt mehrere Aspekte bei der Umwandlung, die Sie berücksichtigen müssen, z. B. was mit Elementen geschehen soll, die nicht umgewandelt werden, und wie Elemente abgebildet werden, die auf eine bestimmte Weise umgewandelt werden sollen. Weitere Informationen zur Umwandlung eines Quellschemas finden Sie unter [Konvertieren von Datenbankschemas mithilfe von AWS SCT](#).

Wenn Sie ein Data Warehouse-Schema umwandeln, müssen vor der Umwandlung ebenfalls einige Dinge berücksichtigt werden. Weitere Informationen finden Sie unter [Konvertierung von Data Warehouse-Schemas in Amazon Redshift mithilfe von AWS SCT](#).

8. Anwenden der Schemaumwandlung auf Ihr Ziel. Weitere Informationen zur Anwendung einer Quellschemaumwandlung finden Sie unter [Den konvertierten Code anwenden](#).
9. Sie können auch verwenden AWS SCT um gespeicherte SQL-Prozeduren und anderen Anwendungscode zu konvertieren. Weitere Informationen finden Sie unter [Anwendungs-SQL konvertieren mit AWS SCT](#)

Sie können auch AWS SCT verwenden, um Ihre Daten aus einer Quell-Datenbank in eine von Amazon verwaltete Datenbank zu migrieren. Beispiele finden Sie unter [Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift](#).

Quellen für AWS SCT

AWS Schema Conversion Tool (AWS SCT) kann Schemas aus den folgenden Quelldatenbanken und -Data Warehouses in eine Zieldatenbank oder ein Ziel-Data Warehouse umwandeln. Informationen zu Berechtigungen, Verbindungen und dazu, was für die Verwendung mit der Zieldatenbank oder dem Data Warehouse konvertiert werden AWS SCT kann, finden Sie in den folgenden Themen.

Informationen zur Verschlüsselung

[Verschlüsselung von Amazon RDS-Verbindungen](#)

Datenbank-Quellen

- [Apache Cassandra als Quelle verwenden](#)
- [Azure SQL Database als Quelle verwenden](#)
- [IBM Db2 for z/OS als Quelle verwenden](#)
- [Verwenden von IBM Db2 LUW als Quelle](#)
- [Verwenden von MySQL als Quelle](#)
- [Oracle Database als Quelle verwenden](#)
- [Verwenden von PostgreSQL als Quelle](#)
- [SAP ASE \(Sybase ASE\) als Quelle verwenden](#)
- [Verwenden von SQL Server als Quelle](#)

Data Warehouse-Quellen

- [Amazon Redshift als Quelle verwenden](#)
- [Azure Synapse Analytics als Quelle verwenden](#)
- [Verwenden von BigQuery als Quelle](#)
- [Greenplum Database als Quelle verwenden](#)
- [Netezza als Quelle verwenden](#)
- [Oracle Data Warehouse als Quelle verwenden](#)
- [Snowflake als Quelle verwenden](#)
- [SQL Server Data Warehouse als Quelle verwenden](#)

- [Teradata als Quelle verwenden](#)
- [Vertica als Quelle verwenden](#)

Große Datenquellen

- [Apache Hadoop als Quelle verwenden](#)
- [Apache Oozie als Quelle verwenden](#)

Verschlüsselung von Amazon RDS- und Amazon Aurora-Verbindungen in AWS SCT

Um verschlüsselte Verbindungen zu Amazon RDS- oder Amazon Aurora-Datenbanken von einer Anwendung aus zu öffnen, müssen Sie AWS Stammzertifikate in eine Form von Schlüsselspeicher importieren. Sie können die Stammzertifikate AWS unter [Verwenden von SSL/TLS zur Verschlüsselung einer Verbindung zu einer DB-Instance](#) im Amazon RDS-Benutzerhandbuch herunterladen.

Zwei Optionen sind verfügbar: ein Stammzertifikat, das für alle AWS Regionen funktioniert, und ein Zertifikatspaket, das sowohl das alte als auch das neue Stammzertifikat enthält.

Je nachdem, welches Sie verwenden möchten, folgen Sie den Schritten in einem der beiden folgenden Verfahren.


Um das Zertifikat oder die Zertifikate in den Windows-Systemspeicher zu importieren

1. Laden Sie ein oder mehrere Zertifikate von einer der folgenden Quellen herunter:

Informationen zum Herunterladen von Zertifikaten finden Sie unter [Verwenden von SSL/TLS zur Verschlüsselung einer Verbindung zu einer DB-Instance](#) im Amazon RDS-Benutzerhandbuch.

2. Geben Sie in Ihrem Windows-Suchfenster ein **Manage computer certificates**. Wenn Sie gefragt werden, ob die Anwendung Änderungen an Ihrem Computer vornehmen lassen soll, wählen Sie Ja.
3. Wenn das Zertifikatsfenster geöffnet wird, erweitern Sie bei Bedarf Zertifikate — Lokaler Computer, damit Sie die Liste der Zertifikate sehen können. Öffnen Sie das Kontextmenü (Rechtsklick) für Trusted Root Certification Authorities und wählen Sie dann Alle Aufgaben, Import.

4. Wählen Sie Weiter und dann Durchsuchen und suchen Sie nach der *.pem Datei, die Sie in Schritt 1 heruntergeladen haben. Wählen Sie Öffnen, um die Zertifikatsdatei auszuwählen, klicken Sie auf Weiter und dann auf Fertig stellen.

 Note

Um die Datei zu finden, ändern Sie den Dateityp im Suchfenster in Alle Dateien (*.*), da dies keine Standardzertifikatserweiterung .pem ist.

5. Erweitern Sie in der Microsoft Management Console die Option Zertifikate. Erweitern Sie dann Trusted Root Certification Authorities, wählen Sie Zertifikate aus und suchen Sie nach dem Zertifikat, um zu bestätigen, dass es existiert. Der Name des Zertifikats beginnt mit Amazon RDS.
6. Starten Sie Ihren Computer neu.

Um das Zertifikat oder die Zertifikate in Java zu importieren KeyStore

1. Laden Sie das Zertifikat oder die Zertifikate von einer der folgenden Quellen herunter:

Informationen zum Herunterladen von Zertifikaten finden Sie unter [Verwenden von SSL/TLS zur Verschlüsselung einer Verbindung zu einer DB-Instance](#) im Amazon RDS-Benutzerhandbuch.

2. Wenn Sie das Zertifikatspaket heruntergeladen haben, teilen Sie es in einzelne Zertifikatsdateien auf. Platzieren Sie dazu jeden Zertifikatsblock, beginnend mit -----BEGIN CERTIFICATE----- und endend mit, -----END CERTIFICATE----- in einer separaten *.pem Datei. Nachdem Sie für jedes Zertifikat eine separate *.pem Datei erstellt haben, können Sie die Zertifikatspaketdatei problemlos entfernen.
3. Öffnen Sie ein Befehlsfenster oder eine Terminalsitzung in dem Verzeichnis, in das Sie das Zertifikat heruntergeladen haben, und führen Sie den folgenden Befehl für jede *.pem Datei aus, die Sie im vorherigen Schritt erstellt haben.

```
keytool -importcert -file <filename>.pem -alias <filename>.pem -keystore storename
```

Example

Im folgenden Beispiel wird davon ausgegangen, dass Sie die eu-west-1-bundle.pem Datei heruntergeladen haben.

```
keytool -importcert -file eu-west-1-bundle.pem -alias eu-west-1-bundle.pem -
keystore trust-2019.ks
Picked up JAVA_TOOL_OPTIONS: -Dlog4j2.formatMsgNoLookups=true
Enter keystore password:
Re-enter new password:
Owner: CN=Amazon RDS Root 2019 CA, OU=Amazon RDS, O="Amazon Web Services, Inc.",
  ST=Washington, L=Seattle, C=US
Issuer: CN=Amazon RDS Root 2019 CA, OU=Amazon RDS, O="Amazon Web Services, Inc.",
  ST=Washington, L=Seattle, C=US
Serial number: c73467369250ae75
Valid from: Thu Aug 22 19:08:50 CEST 2019 until: Thu Aug 22 19:08:50 CEST 2024
Certificate fingerprints:
    SHA1: D4:0D:DB:29:E3:75:0D:FF:A6:71:C3:14:0B:BF:5F:47:8D:1C:80:96
    SHA256:
    F2:54:C7:D5:E9:23:B5:B7:51:0C:D7:9E:F7:77:7C:1C:A7:E6:4A:3C:97:22:E4:0D:64:54:78:FC:70:AA:
Signature algorithm name: SHA256withRSA
Subject Public Key Algorithm: 2048-bit RSA key
Version: 3

Extensions:

#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 73 5F 60 D8 BC CB 03 98   F4 2B 17 34 2E 36 5A A6   s_`.....+.4.6Z.
0010: 60 FF BC 1F                   `...
]
]

#2: ObjectId: 2.5.29.19 Criticality=true
BasicConstraints:[
  CA:true
  PathLen:2147483647
]

#3: ObjectId: 2.5.29.15 Criticality=true
KeyUsage [
  Key_CertSign
  Crl_Sign
]

#4: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
```

```

KeyIdentifier [
0000: 73 5F 60 D8 BC CB 03 98   F4 2B 17 34 2E 36 5A A6   s_`.....+.4.6Z.
0010: 60 FF BC 1F               `...
]
]

Trust this certificate? [no]:  yes
Certificate was added to keystore

```

4. Fügen Sie den Keystore als Truststore hinzu. AWS SCT Wählen Sie dazu im Hauptmenü Einstellungen, Allgemeine Einstellungen, Sicherheit, Trust Store und dann Select existing Trust Store aus.

Nachdem Sie den Trust Store hinzugefügt haben, können Sie ihn verwenden, um eine SSL-fähige Verbindung zu konfigurieren, wenn Sie eine AWS SCT Verbindung zur Datenbank herstellen. Wählen AWS SCT Sie im Dialogfeld Mit Datenbank verbinden die Option SSL verwenden und wählen Sie den zuvor eingegebenen Trust Store aus.

Verwendung von Apache Cassandra als Quelle für AWS SCT

Sie können AWS SCT Keyspaces von Apache Cassandra nach Amazon DynamoDB konvertieren.

Verbindung zu Apache Cassandra als Quelle herstellen

Gehen Sie wie folgt vor, um mit dem eine Verbindung zu Ihrer Apache Cassandra-Quelldatenbank herzustellen. AWS Schema Conversion Tool

So stellen Sie eine Verbindung zu einer Apache Cassandra-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Cassandra und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankanmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:

- Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen der Apache Cassandra-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
User name und Password	<p>Geben Sie die Datenbankmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>

Parameter	Action
Use SSL	Wählen Sie diese Option, wenn Sie Secure Sockets Layer (SSL) verwenden möchten, um eine Verbindung zu Ihrer Datenbank herzustellen. Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an: <ul style="list-style-type: none"> Trust Store: Der zu verwendende Trust Store. Schlüsselspeicher: Der zu verwendende Schlüssel speicher.
Passwort speichern	AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.

- Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
- Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Verwendung von Apache Hadoop als Quelle für AWS SCT

Sie können die AWS SCT Befehlszeilenschnittstelle (CLI) verwenden, um von Apache Hadoop zu Amazon EMR zu migrieren. AWS SCT verwendet Ihren Amazon S3-Bucket als temporären Speicher für Ihre Daten während der Migration.

AWS SCT unterstützt als Quelle Apache Hadoop Version 2.2.0 und höher. AWS SCT unterstützt auch Apache Hive Version 0.13.0 und höher.

AWS SCT unterstützt als Ziel Amazon EMR Version 6.3.0 und höher. AWS SCT unterstützt außerdem als Ziel Apache Hadoop Version 2.6.0 und höher sowie Apache Hive Version 0.13.0 und höher.

Themen

- [Voraussetzungen für die Verwendung von Apache Hadoop als Quelle](#)
- [Berechtigungen für die Verwendung von Hive als Quelle](#)

- [Berechtigungen für die Verwendung von HDFS als Quelle](#)
- [Berechtigungen für die Verwendung von HDFS als Ziel](#)
- [Verbindung zu Apache Hadoop als Quelle herstellen](#)
- [Verbindung zu Ihren Hive- und HDFS-Quelldiensten herstellen](#)
- [Verbindung zu Amazon EMR als Ziel herstellen](#)

Voraussetzungen für die Verwendung von Apache Hadoop als Quelle

Die folgenden Voraussetzungen sind erforderlich, um mit der AWS SCT CLI eine Verbindung zu Apache Hadoop herzustellen.

- Erstellen Sie einen Amazon S3-Bucket, um Daten während der Migration zu speichern. Anschließend können Sie Daten nach Amazon EMR HDFS kopieren oder Amazon S3 als Datenrepository für Ihre Hadoop-Workloads verwenden. Weitere Informationen finden Sie unter [Erstellen eines Buckets](#) im Amazon S3-Benutzerhandbuch.
- Erstellen Sie mit der AmazonS3FullAccess Richtlinie eine AWS Identity and Access Management (IAM-) Rolle. AWS SCT verwendet diese IAM-Rolle, um auf Ihren Amazon S3-Bucket zuzugreifen.
- Notieren Sie sich Ihren AWS geheimen Schlüssel und Ihren AWS geheimen Zugangsschlüssel. Weitere Informationen zu AWS Zugriffsschlüsseln finden Sie unter [Verwaltung von Zugriffsschlüsseln](#) im IAM-Benutzerhandbuch.
- Erstellen und konfigurieren Sie einen Amazon EMR-Zielcluster. Weitere Informationen finden Sie unter [Erste Schritte mit Amazon EMR](#) im Amazon EMR Management Guide.
- Installieren Sie das `distcp` Utility auf Ihrem Apache Hadoop-Quellcluster. Installieren Sie das `s3-dist-cp` Dienstprogramm außerdem auf Ihrem Amazon EMR-Zielcluster. Stellen Sie sicher, dass Ihre Datenbankbenutzer berechtigt sind, diese Dienstprogramme auszuführen.
- Konfigurieren Sie die `core-site.xml` Datei in Ihrem Hadoop-Quellcluster für die Verwendung des s3a-Protokolls. Stellen Sie dazu den `fs.s3a.aws.credentials.provider` Parameter auf einen der folgenden Werte ein.
 - `org.apache.hadoop.fs.s3a.TemporaryAWSCredentialsProvider`
 - `org.apache.hadoop.fs.s3a.SimpleAWSCredentialsProvider`
 - `org.apache.hadoop.fs.s3a.AnonymousAWSCredentialsProvider`
 - `org.apache.hadoop.fs.s3a.auth.AssumedRoleCredentialProvider`

Sie können das folgende Codebeispiel in die `core-site.xml` Datei einfügen.

```
<property>
  <name>fs.s3a.aws.credentials.provider</name>
  <value>org.apache.hadoop.fs.s3a.SimpleAWSCredentialsProvider</value>
</property>
```

Das vorherige Beispiel zeigt eine der vier Optionen aus der vorherigen Optionsliste. Wenn Sie den `fs.s3a.aws.credentials.provider` Parameter nicht in der `core-site.xml` Datei festlegen, wird AWS SCT der Anbieter automatisch ausgewählt.

Berechtigungen für die Verwendung von Hive als Quelle

Die für einen Hive-Quellbenutzer erforderlichen Berechtigungen lauten wie folgt:

- READZugriff auf die Quelldatenordner und auf den Amazon S3-Quell-Bucket
- READ+WRITEZugriff auf die Amazon S3-Zwischenbuckets und Ziel-Buckets

Um die Migrationsgeschwindigkeit zu erhöhen, empfehlen wir, die Komprimierung für ACID-Transaktionsquelltabellen durchzuführen.

Die für einen Amazon EMR Hive-Zielbenutzer erforderlichen Berechtigungen lauten wie folgt:

- READZugriff auf den Amazon S3-Ziel-Bucket
- READ+WRITEZugriff auf den Amazon S3-Zwischenbucket
- READ+WRITEZugriff auf die HDFS-Zielordner

Berechtigungen für die Verwendung von HDFS als Quelle

Die für HDFS als Quelle erforderlichen Berechtigungen lauten wie folgt:

- EXECUTE für die Region NameNode
- EXECUTE+READfür alle Quellordner und Dateien, die Sie in das Migrationsprojekt aufnehmen
- READ+WRITEfür das `tmp` Verzeichnis in der NameNode, um Spark-Jobs auszuführen und Dateien vor der Migration zu Amazon S3 zu speichern

In HDFS ist für alle Operationen ein Traversalzugriff erforderlich. Der Traversalzugriff erfordert die EXECUTE Erlaubnis für alle vorhandenen Komponenten des Pfads, mit Ausnahme der letzten Pfadkomponente. Beispielsweise muss Ihr Benutzer für jeden Vorgang/foo/bar/baz, auf den zugegriffen wird, über die EXECUTE Berechtigung für //foo, und verfügen/foo/bar.

Das folgende Codebeispiel zeigt, wie Sie EXECUTE+READ Berechtigungen für Ihre Quellordner und -dateien sowie READ+WRITE Berechtigungen für das tmp Verzeichnis gewähren.

```
hadoop fs -chmod -R 744 /user/hdfs-data
hadoop fs -chmod -R 766 /tmp
```

Berechtigungen für die Verwendung von HDFS als Ziel

Die für Amazon EMR HDFS als Ziel erforderlichen Berechtigungen lauten wie folgt:

- EXECUTE für den NameNode des Amazon EMR-Ziel-Clusters
- READ+WRITE für die HDFS-Zielordner, in denen Sie Daten nach der Migration speichern werden

Verbindung zu Apache Hadoop als Quelle herstellen

Sie können Apache Hadoop als Quelle in AWS SCT Version 1.0.670 oder höher verwenden. Sie können Hadoop-Cluster nur über die AWS SCT Befehlszeilenschnittstelle (CLI) zu Amazon EMR migrieren. Bevor Sie beginnen, machen Sie sich mit der Befehlszeilenschnittstelle von vertraut AWS SCT. Weitere Informationen finden Sie unter [AWS SCT CLI-Referenz](#).

So stellen Sie in der CLI eine Verbindung zu Apache Hadoop her AWS SCT

1. Erstellen Sie ein neues AWS SCT CLI-Skript oder bearbeiten Sie eine vorhandene Szenariovorlage. Sie können die HadoopMigrationTemplate.scts Vorlage beispielsweise herunterladen und bearbeiten. Weitere Informationen finden Sie unter [CLI-Szenarien abrufen](#).
2. Konfigurieren Sie die AWS SCT Anwendungseinstellungen wie den Treiberspeicherort und den Protokollordner.

Laden Sie den erforderlichen JDBC-Treiber herunter und geben Sie den Speicherort an, an dem Sie die Datei speichern. Weitere Informationen finden Sie unter [Die erforderlichen Datenbanktreiber werden heruntergeladen](#).

Das folgende Codebeispiel zeigt Ihnen, wie Sie den Pfad zum Apache Hive-Treiber hinzufügen. Nachdem Sie dieses Codebeispiel ausgeführt haben, AWS SCT werden Protokolldateien im c : \sct Ordner gespeichert.

```
SetGlobalSettings
  -save: 'true'
  -settings: '{
    "hive_driver_file": "c:\\sct\\HiveJDBC42.jar",
    "log_folder": "c:\\sct",
    "console_log_folder": "c:\\sct"
  }'
/
```

Sie können dieses Beispiel und die folgenden Beispiele in Windows verwenden.

3. Erstellen Sie ein neues AWS SCT Projekt.

Im folgenden Codebeispiel wird das `hadoop_emr` Projekt in dem c : \sct Ordner erstellt.

```
CreateProject
  -name: 'hadoop_emr'
  -directory: 'c:\sct'
/
```

4. Fügen Sie Ihren Quell-Hadoop-Cluster zum Projekt hinzu.

Verwenden Sie den `AddSourceCluster` Befehl, um eine Verbindung zum Hadoop-Quellcluster herzustellen. Stellen Sie sicher, dass Sie Werte für die folgenden obligatorischen Parameter angeben: `namehost`, `port`, und `user`. Andere Parameter sind optional.

Das folgende Codebeispiel fügt den Quell-Hadoop-Cluster hinzu. In diesem Beispiel `HADOOP_SOURCE` wird der Name des Quellclusters festgelegt. Verwenden Sie diesen Objektnamen, um Hive- und HDFS-Dienste zum Projekt hinzuzufügen und Zuordnungsregeln zu erstellen.

```
AddSourceCluster
  -name: 'HADOOP_SOURCE'
  -vendor: 'HADOOP'
  -host: 'hadoop_address'
  -port: '22'
  -user: 'hadoop_user'
```

```
-password: 'hadoop_password'  
-useSSL: 'true'  
-privateKeyPath: 'c:\path\name.pem'  
-passPhrase: 'hadoop_passphrase'
```

/

Ersetzen Sie im vorherigen Beispiel *hadoop_address* durch die IP-Adresse Ihres Hadoop-Clusters. Konfigurieren Sie bei Bedarf den Wert der Portoption. Als Nächstes ersetzen Sie *hadoop_user* und *hadoop_password* durch den Namen Ihres Hadoop-Benutzers und das Passwort für diesen Benutzer. Geben Sie für *path\ name* den Namen und den Pfad zur PEM-Datei für Ihren Hadoop-Quellcluster ein.

5. Speichern Sie Ihr CLI-Skript. Fügen Sie als Nächstes die Verbindungsinformationen für Ihre Hive- und HDFS-Dienste hinzu.

Verbindung zu Ihren Hive- und HDFS-Quelldiensten herstellen

Sie können mit der CLI eine Verbindung zu Ihren Hive- und HDFS-Quelldiensten herstellen. AWS SCT Um eine Verbindung zu Apache Hive herzustellen, verwenden Sie den Hive JDBC-Treiber Version 2.3.4 oder höher. Weitere Informationen finden Sie unter [Die erforderlichen Datenbanktreiber werden heruntergeladen](#).

AWS SCT stellt mit dem `hadoop` Cluster-Benutzer eine Verbindung zu Apache Hive her. Verwenden Sie dazu die `AddSourceClusterHDFS` Befehle `AddSourceClusterHive` und. Sie können einen der folgenden Ansätze verwenden.

- Erstellen Sie einen neuen SSH-Tunnel.

Geben Sie unter `createTunnel` den Wert **true** ein. Geben Sie für `host` die interne IP-Adresse Ihres Quell-Hive- oder HDFS-Dienstes ein. Geben Sie für `port` den Serviceport Ihres Hive- oder HDFS-Dienstes ein.

Geben Sie als Nächstes Ihre Hive- oder HDFS-Anmeldeinformationen für `user` und ein. `password` Weitere Informationen zu SSH-Tunneln finden Sie unter [Einrichten eines SSH-Tunnels zum primären Knoten mithilfe der lokalen Portweiterleitung](#) im Amazon EMR Management Guide.

- Verwenden Sie einen vorhandenen SSH-Tunnel.

Geben Sie unter `host` den Wert **localhost** ein. Geben Sie für `port` den lokalen Port aus den SSH-Tunnelparametern ein.

- Stellen Sie eine direkte Verbindung zu Ihren Hive- und HDFS-Diensten her.

Geben Sie für `host` die IP-Adresse oder den Hostnamen Ihres Quell-Hive- oder HDFS-Dienstes ein. Geben Sie für `port` den Serviceport Ihres Hive- oder HDFS-Dienstes ein. Geben Sie als Nächstes Ihre Hive- oder HDFS-Anmeldeinformationen für `user` und ein `password`

So stellen Sie in der CLI eine Verbindung zu Hive und HDFS her AWS SCT

1. Öffnen Sie Ihr CLI-Skript, das die Verbindungsinformationen für Ihren Hadoop-Quellcluster enthält. Stellen Sie sicher, dass Sie den Namen des Hadoop-Clusters verwenden, den Sie im vorherigen Schritt definiert haben.
2. Fügen Sie dem Projekt Ihren Hive-Quelldienst hinzu.

Verwenden Sie den `AddSourceClusterHive` Befehl, um den Quell-Hive-Dienst zu verbinden. Stellen Sie sicher, dass Sie Werte für die folgenden obligatorischen Parameter angeben: `userpassword`, `cluster`, `name`, und `port`. Andere Parameter sind optional.

Das folgende Codebeispiel erstellt einen Tunnel für AWS SCT die Arbeit mit Ihrem Hive-Dienst. Dieser Quell-Hive-Dienst läuft auf demselben PC wie AWS SCT. In diesem Beispiel wird der `HADOOP_SOURCE` Quellcluster aus dem vorherigen Beispiel verwendet.

```
AddSourceClusterHive
  -cluster: 'HADOOP_SOURCE'
  -name: 'HIVE_SOURCE'
  -host: 'localhost'
  -port: '10005'
  -user: 'hive_user'
  -password: 'hive_password'
  -createTunnel: 'true'
  -localPort: '10005'
  -remoteHost: 'hive_remote_address'
  -remotePort: 'hive_port'
/
```

Das folgende Codebeispiel stellt eine Verbindung zu Ihrem Hive-Dienst ohne Tunnel her.

```
AddSourceClusterHive
  -cluster: 'HADOOP_SOURCE'
  -name: 'HIVE_SOURCE'
  -host: 'hive_address'
```

```
-port: 'hive_port'  
-user: 'hive_user'  
-password: 'hive_password'
```

/

Ersetzen Sie in den vorherigen Beispielen *hive_user* und *hive_password* durch den Namen Ihres Hive-Benutzers und das Passwort für diesen Benutzer.

Als Nächstes ersetzen Sie *hive_address* und *hive_port* durch die *NameNode IP-Adresse und den Port* Ihres Hadoop-Quellclusters.

Für *hive_remote_address* können Sie den Standardwert `127.0.0.1` oder die NameNode IP-Adresse Ihres Hive-Quelldienstes verwenden.

3. Fügen Sie dem Projekt Ihren HDFS-Quelldienst hinzu.

Verwenden Sie den `AddSourceClusterHDFS` Befehl, um den Quell-HDFS-Dienst zu verbinden. Stellen Sie sicher, dass Sie Werte für die folgenden obligatorischen Parameter angeben: `userpassword`, `clustername`, und `port`. Andere Parameter sind optional.

Stellen Sie sicher, dass Ihr Benutzer über die erforderlichen Berechtigungen verfügt, um Daten aus Ihrem HDFS-Quelldienst zu migrieren. Weitere Informationen finden Sie unter [Berechtigungen für die Verwendung von Hive als Quelle](#).

Das folgende Codebeispiel erstellt einen Tunnel für AWS SCT die Arbeit mit Ihrem Apache HDFS-Dienst. In diesem Beispiel wird der `HADOOP_SOURCE` Quellcluster verwendet, den Sie zuvor erstellt haben.

```
AddSourceClusterHDFS  
-cluster: 'HADOOP_SOURCE'  
-name: 'HDFS_SOURCE'  
-host: 'localhost'  
-port: '9005'  
-user: 'hdfs_user'  
-password: 'hdfs_password'  
-createTunnel: 'true'  
-localPort: '9005'  
-remoteHost: 'hdfs_remote_address'  
-remotePort: 'hdfs_port'
```

/

Der folgende Code stellt eine Verbindung zu Ihrem Apache HDFS-Dienst ohne Tunnel her.

```
AddSourceClusterHDFS
  -cluster: 'HADOOP_SOURCE'
  -name: 'HDFS_SOURCE'
  -host: 'hdfs_address'
  -port: 'hdfs_port'
  -user: 'hdfs_user'
  -password: 'hdfs_password'
/
```

Ersetzen Sie in den vorherigen Beispielen *hdfs_user* und *hdfs_password* durch den Namen Ihres HDFS-Benutzers und das Passwort für diesen Benutzer.

Als Nächstes ersetzen Sie *hdfs_address* und *hdfs_port* durch die *NameNode IP-Adresse und den Port* Ihres Hadoop-Quellclusters.

Für *hdfs_remote_address* können Sie den Standardwert `127.0.0.1` oder die NameNode IP-Adresse Ihres Hive-Quelldienstes verwenden.

4. Speichern Sie Ihr CLI-Skript. Fügen Sie als Nächstes die Verbindungsinformationen für Ihren Amazon EMR-Zielcluster und die Migrationsbefehle hinzu.

Verbindung zu Amazon EMR als Ziel herstellen

Sie können mit der AWS SCT CLI eine Verbindung zu Ihrem Amazon EMR-Zielcluster herstellen. Dazu autorisieren Sie eingehenden Datenverkehr und verwenden SSH. In diesem Fall AWS SCT verfügt er über alle erforderlichen Berechtigungen, um mit Ihrem Amazon EMR-Cluster zu arbeiten. Weitere Informationen finden Sie unter [Bevor Sie mit SSH eine Verbindung zum primären Knoten herstellen und eine Verbindung zum primären Knoten herstellen](#) im Amazon EMR Management Guide.

AWS SCT stellt mit dem hadoop Cluster-Benutzer eine Verbindung zu Amazon EMR Hive her. Um eine Verbindung zu Amazon EMR Hive herzustellen, verwenden Sie den Hive JDBC-Treiber Version 2.6.2.1002 oder höher. Weitere Informationen finden Sie unter [Die erforderlichen Datenbanktreiber werden heruntergeladen](#).

So stellen Sie in der CLI eine Verbindung zu Amazon EMR her AWS SCT

1. Öffnen Sie Ihr CLI-Skript, das die Verbindungsinformationen für Ihren Hadoop-Quellcluster enthält. Fügen Sie die Amazon EMR-Zielanmeldeinformationen zu dieser Datei hinzu.
2. Fügen Sie Ihren Amazon EMR-Zielcluster zum Projekt hinzu.

Das folgende Codebeispiel fügt den Amazon EMR-Zielcluster hinzu. In diesem Beispiel HADOOP_TARGET wird der Name des Zielclusters festgelegt. Verwenden Sie diesen Objektnamen, um Ihre Hive- und HDFS-Dienste sowie einen Amazon S3-Bucket-Ordner zum Projekt hinzuzufügen und Zuordnungsregeln zu erstellen.

```
AddTargetCluster
  -name: 'HADOOP_TARGET'
  -vendor: 'AMAZON_EMR'
  -host: 'ec2-44-44-55-66.eu-west-1.EXAMPLE.amazonaws.com'
  -port: '22'
  -user: 'emr_user'
  -password: 'emr_password'
  -useSSL: 'true'
  -privateKeyPath: 'c:\path\name.pem'
  -passPhrase: '1234567890abcdef0!'
  -s3Name: 'S3_TARGET'
  -accessKey: 'AKIAIOSFODNN7EXAMPLE'
  -secretKey: 'wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY'
  -region: 'eu-west-1'
  -s3Path: 'doc-example-bucket/example-folder'
```

/

Geben Sie im vorherigen Beispiel Ihre AWS Ressourcennamen und Amazon EMR-Verbindungsinformationen ein. Dazu gehören die IP-Adresse Ihres Amazon EMR-Clusters, der AWS Zugriffsschlüssel, der AWS geheime Zugriffsschlüssel und der Amazon S3-Bucket. Konfigurieren Sie bei Bedarf den Wert der Portvariablen. Als Nächstes ersetzen Sie *emr_user* und *emr_password* durch den Namen Ihres Amazon EMR-Benutzers und das Passwort für diesen Benutzer. Geben Sie für *path\ name* den Namen und den Pfad zur PEM-Datei für Ihren Amazon EMR-Zielcluster ein. Weitere Informationen finden [Sie unter PEM-Datei für den EMR-Clusterzugriff herunterladen](#).

3. Fügen Sie Ihren Amazon S3-Ziel-Bucket zum Projekt hinzu.

Das folgende Codebeispiel fügt den Amazon S3-Ziel-Bucket hinzu. In diesem Beispiel wird der HADOOP_TARGET Cluster verwendet, den Sie zuvor erstellt haben.

```
AddTargetClusterS3
  -cluster: 'HADOOP_TARGET'
  -Name: 'S3_TARGET'
  -accessKey: 'AKIAIOSFODNN7EXAMPLE'
  -secretKey: 'wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY'
  -region: 'eu-west-1'
  -s3Path: 'doc-example-bucket/example-folder'
/
```

Geben Sie im vorherigen Beispiel Ihren AWS Zugriffsschlüssel, Ihren AWS geheimen Zugriffsschlüssel und Ihren Amazon S3-Bucket ein.

4. Fügen Sie Ihren Ziel-Hive-Dienst zum Projekt hinzu.

Das folgende Codebeispiel erstellt einen Tunnel für AWS SCT die Arbeit mit Ihrem Ziel-Hive-Dienst. In diesem Beispiel wird der HADOOP_TARGET Zielcluster verwendet, den Sie zuvor erstellt haben.

```
AddTargetClusterHive
  -cluster: 'HADOOP_TARGET'
  -name: 'HIVE_TARGET'
  -host: 'localhost'
  -port: '10006'
  -user: 'hive_user'
  -password: 'hive_password'
  -createTunnel: 'true'
  -localPort: '10006'
  -remoteHost: 'hive_address'
  -remotePort: 'hive_port'
/
```

Ersetzen Sie im vorherigen Beispiel *hive_user* und *hive_password* durch den Namen Ihres Hive-Benutzers und das Passwort für diesen Benutzer.

Ersetzen Sie anschließend *hive_address* durch den Standardwert `127.0.0.1` oder durch die NameNode IP-Adresse Ihres Ziel-Hive-Dienstes. Als Nächstes ersetzen Sie *hive_port* durch *den Port* Ihres Ziel-Hive-Dienstes.

5. Fügen Sie dem Projekt Ihren Ziel-HDFS-Dienst hinzu.

Das folgende Codebeispiel erstellt einen Tunnel für AWS SCT die Arbeit mit Ihrem Apache HDFS-Dienst. In diesem Beispiel wird der HADOOP_TARGET Zielcluster verwendet, den Sie zuvor erstellt haben.

```
AddTargetClusterHDFS
  -cluster: 'HADOOP_TARGET'
  -name: 'HDFS_TARGET'
  -host: 'localhost'
  -port: '8025'
  -user: 'hdfs_user'
  -password: 'hdfs_password'
  -createTunnel: 'true'
  -localPort: '8025'
  -remoteHost: 'hdfs_address'
  -remotePort: 'hdfs_port'
/
```

Ersetzen Sie im vorherigen Beispiel *hdfs_user* und *hdfs_password* durch den Namen Ihres HDFS-Benutzers und das Passwort für diesen Benutzer.

Als Nächstes ersetzen Sie *hdfs_address* und *hdfs_port* durch die private IP-Adresse und den Port des NameNode HDFS-Zieldienstes.

6. Speichern Sie Ihr CLI-Skript. Fügen Sie als Nächstes Zuordnungsregeln und Migrationsbefehle hinzu. Weitere Informationen finden Sie unter [Migration von Apache Hadoop zu Amazon EMR](#).

Verwendung von Apache Oozie als Quelle für AWS SCT

Sie können die AWS SCT Befehlszeilenschnittstelle (CLI) verwenden, um Apache Oozie-Workflows zu konvertieren. AWS Step Functions Nachdem Sie Ihre Apache Hadoop-Workloads zu Amazon EMR migriert haben, können Sie einen nativen Service in der verwenden, um Ihre Jobs AWS Cloud zu orchestrieren. Weitere Informationen finden Sie unter [Apache Hadoop als Quelle verwenden](#).

AWS SCTkonvertiert Ihre Oozie-Workflows in Funktionen, die AWS Step Functions nicht unterstützt werden, AWS Step Functions und verwendet AWS Lambda sie, um sie zu emulieren. AWS SCTKonvertiert außerdem Ihre Oozie-Jobeigenschaften in. AWS Systems Manager

Um Apache Oozie-Workflows zu konvertieren, stellen Sie sicher, dass Sie AWS SCT Version 1.0.671 oder höher verwenden. Machen Sie sich auch mit der Befehlszeilenschnittstelle von vertraut AWS SCT. Weitere Informationen finden Sie unter [AWS SCT CLI-Referenz](#).

Voraussetzungen für die Verwendung von Apache Oozie als Quelle

Die folgenden Voraussetzungen sind erforderlich, um mit der AWS SCT CLI eine Verbindung zu Apache Oozie herzustellen.

- Erstellen Sie einen Amazon S3-Bucket, um die Definitionen von State Machines zu speichern. Sie können diese Definitionen verwenden, um Ihre State-Maschinen zu konfigurieren. Weitere Informationen finden Sie unter [Erstellen eines Buckets](#) im Amazon S3-Benutzerhandbuch.
- Erstellen Sie mit der AmazonS3FullAccess Richtlinie eine AWS Identity and Access Management (IAM-) Rolle. AWS SCT verwendet diese IAM-Rolle, um auf Ihren Amazon S3-Bucket zuzugreifen.
- Notieren Sie sich Ihren AWS geheimen Schlüssel und Ihren AWS geheimen Zugangsschlüssel. Weitere Informationen zu AWS Zugriffsschlüsseln finden Sie unter [Verwaltung von Zugriffsschlüsseln](#) im IAM-Benutzerhandbuch.
- Speichern Sie Ihre AWS Anmeldeinformationen und die Informationen zu Ihrem Amazon S3-Bucket im AWS Serviceprofil in den globalen Anwendungseinstellungen. Verwenden Sie dann AWS SCT dieses AWS Dienstprofil, um mit Ihren AWS Ressourcen zu arbeiten. Weitere Informationen finden Sie unter [Speichern von AWS Serviceprofilen in AWS SCT](#).

Um mit Ihren Apache Oozie-Quellworkflows arbeiten zu können, AWS SCT ist die spezifische Struktur Ihrer Quelldateien erforderlich. Jeder Ihrer Bewerbungsordner muss die `job.properties` Datei enthalten. Diese Datei enthält Schlüssel-Wert-Paare Ihrer Jobeigenschaften. Außerdem muss jeder Ihrer Bewerbungsordner die `workflow.xml` Datei enthalten. Diese Datei beschreibt die Aktionsknoten und Kontrollflussknoten Ihres Workflows.

Verbindung zu Apache Oozie als Quelle herstellen

Gehen Sie wie folgt vor, um eine Verbindung zu Ihren Apache Oozie-Quelldateien herzustellen.

So stellen Sie in der CLI eine Verbindung zu Apache Oozie her AWS SCT

1. Erstellen Sie ein neues AWS SCT CLI-Skript oder bearbeiten Sie eine vorhandene Szenariovorlage. Sie können die `OozieConversionTemplate.scts` Vorlage beispielsweise herunterladen und bearbeiten. Weitere Informationen finden Sie unter [CLI-Szenarien abrufen](#).

2. Konfigurieren Sie die AWS SCT Anwendungseinstellungen.

Das folgende Codebeispiel speichert die Anwendungseinstellungen und ermöglicht das Speichern von Passwörtern in Ihrem Projekt. Sie können diese gespeicherten Einstellungen in anderen Projekten verwenden.

```
SetGlobalSettings
  -save: 'true'
  -settings: '{
    "store_password": "true"
  }'
/
```

3. Erstellen Sie ein neues AWS SCT Projekt.

Im folgenden Codebeispiel wird das `oozie` Projekt in dem `c:\sct` Ordner erstellt.

```
CreateProject
  -name: 'oozie'
  -directory: 'c:\sct'
/
```

4. Fügen Sie den Ordner mit Ihren Apache Oozie-Quelldateien mit dem `AddSource` Befehl zum Projekt hinzu. Stellen Sie sicher, dass Sie den `APACHE_00ZIE` Wert für den `vendor` Parameter verwenden. Geben Sie außerdem Werte für die folgenden erforderlichen Parameter an: `name` und `mappingsFolder`.

Das folgende Codebeispiel fügt Apache Oozie als Quelle in Ihr AWS SCT Projekt ein. In diesem Beispiel wird ein Quellobjekt mit dem Namen `erstellt00ZIE`. Verwenden Sie diesen Objektname, um Zuordnungsregeln hinzuzufügen. Nachdem Sie dieses Codebeispiel ausgeführt haben, verwenden Sie den `c:\oozie` Ordner, um Ihre Quelldateien in das Projekt zu laden.

```
AddSource
  -name: '00ZIE'
  -vendor: 'APACHE_00ZIE'
  -mappingsFolder: 'c:\oozie'
/
```

Sie können dieses Beispiel und die folgenden Beispiele in Windows verwenden.

5. Stellen Sie mit dem ConnectSource Befehl eine Verbindung zu Ihren Apache Oozie-Quelldateien her. Verwenden Sie den Namen Ihres Quellobjekts, das Sie im vorherigen Schritt definiert haben.

```
ConnectSource
  -name: 'OOZIE'
  -mappingsFolder: 'c:\oozie'
/
```

6. Speichern Sie Ihr CLI-Skript. Fügen Sie als Nächstes die Verbindungsinformationen für Ihren AWS Step Functions Dienst hinzu.

Berechtigungen für die Verwendung von AWS Lambda Funktionen im Erweiterungspaket

AWS SCT erstellt für die Quellfunktionen, die AWS Step Functions nicht unterstützt werden, ein Erweiterungspaket. Dieses Erweiterungspaket enthält AWS Lambda Funktionen, die Ihre Quellfunktionen emulieren.

Um dieses Erweiterungspaket zu verwenden, erstellen Sie eine AWS Identity and Access Management (IAM-) Rolle mit den folgenden Berechtigungen.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "lambda",
      "Effect": "Allow",
      "Action": [
        "lambda:InvokeFunction"
      ],
      "Resource": [
        "arn:aws:lambda:*:498160209112:function:LoadParameterInitialState:*",
        "arn:aws:lambda:*:498160209112:function:EvaluateJSPELExpressions:*"
      ]
    },
    {
      "Sid": "emr",
      "Effect": "Allow",
      "Action": [
        "elasticmapreduce:DescribeStep",
```

```

        "elasticmapreduce:AddJobFlowSteps"
    ],
    "Resource": [
        "arn:aws:elasticmapreduce:*:498160209112:cluster/*"
    ]
},
{
    "Sid": "s3",
    "Effect": "Allow",
    "Action": [
        "s3:GetObject"
    ],
    "Resource": [
        "arn:aws:s3:::*/*"
    ]
}
]
}

```

Um das Erweiterungspaket anzuwenden, AWS SCT ist eine IAM-Rolle mit den folgenden Berechtigungen erforderlich.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:ListRolePolicies",
                "iam:CreateRole",
                "iam:TagRole",
                "iam:PutRolePolicy",
                "iam>DeleteRolePolicy",
                "iam>DeleteRole",
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::ACCOUNT_NUMBER:role/sct/*"
            ]
        },
        {
            "Effect": "Allow",

```



```
        "Action": [
            "iam:GetRole",
            "iam:ListRolePolicies"
        ],
        "Resource": [
            "arn:aws:iam::ACCOUNT_NUMBER:role/
lambda_LoadParameterInitialStateRole",
            "arn:aws:iam::ACCOUNT_NUMBER:role/lambda_EvaluateJSPELExpressionsRole",
            "arn:aws:iam::ACCOUNT_NUMBER:role/
stepFunctions_MigratedOozieWorkflowRole"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "lambda:GetFunction",
            "lambda:CreateFunction",
            "lambda:UpdateFunctionCode",
            "lambda>DeleteFunction"
        ],
        "Resource": [
            "arn:aws:lambda:*:ACCOUNT_NUMBER:function:LoadParameterInitialState",
            "arn:aws:lambda:*:ACCOUNT_NUMBER:function:EvaluateJSPELExpressions"
        ]
    }
]
}
```

AWS Step FunctionsAls Ziel eine Verbindung herstellen

Gehen Sie wie folgt vor, um eine Verbindung zu AWS Step Functions als Ziel herzustellen.

Um AWS Step Functions in der AWS SCT CLI eine Verbindung herzustellen

1. Öffnen Sie Ihr CLI-Skript, das die Verbindungsinformationen für Ihre Apache Oozie-Quelldateien enthält.
2. Fügen Sie mithilfe des `AddTarget` Befehls die Informationen zu Ihrem Migrationsziel in das AWS SCT Projekt ein. Stellen Sie sicher, dass Sie den `STEP_FUNCTIONS` Wert für den `vendor` Parameter verwenden. Geben Sie außerdem Werte für die folgenden erforderlichen Parameter an: `name` und `profile`.

Das folgende Codebeispiel wird Ihrem AWS SCT Projekt AWS Step Functions als Quelle hinzugefügt. In diesem Beispiel wird ein Zielobjekt mit dem Namen `AWS_STEP_FUNCTIONS` erstellt. Verwenden Sie diesen Objektname, wenn Sie Zuordnungsregeln erstellen. In diesem Beispiel wird außerdem ein AWS SCT Dienstprofil verwendet, das Sie im Schritt „Voraussetzungen“ erstellt haben. Stellen Sie sicher, dass Sie *profile_name* durch den Namen Ihres Profils ersetzen.

```
AddTarget
  -name: 'AWS_STEP_FUNCTIONS'
  -vendor: 'STEP_FUNCTIONS'
  -profile: 'profile_name'
/
```

Wenn Sie das AWS Dienstprofil nicht verwenden, stellen Sie sicher, dass Sie Werte für die folgenden erforderlichen Parameter angeben: `accessKeysecretKey`, `awsRegion`, und `s3Path`. Verwenden Sie diese Parameter, um Ihren AWS geheimen Zugriffsschlüssel, Ihren AWS geheimen Schlüssel und den Pfad zu Ihrem Amazon S3-Bucket anzugeben. AWS-Region

3. Stellen Sie AWS Step Functions mithilfe des `ConnectTarget` Befehls eine Verbindung her. Verwenden Sie den Namen Ihres Zielobjekts, das Sie im vorherigen Schritt definiert haben.

Das folgende Codebeispiel stellt mithilfe Ihres AWS Dienstprofils eine Verbindung zum `AWS_STEP_FUNCTIONS` Zielobjekt her. Stellen Sie sicher, dass Sie *profile_name* durch den Namen Ihres Profils ersetzen.

```
ConnectTarget
  -name: 'AWS_STEP_FUNCTIONS'
  -profile: 'profile_name'
/
```

4. Speichern Sie Ihr CLI-Skript. Fügen Sie als Nächstes Zuordnungsregeln und Migrationsbefehle hinzu. Weitere Informationen finden Sie unter [Konvertierung von Apache Oozie zu AWS Step Functions](#).

Verwenden von Azure SQL Database als Quelle für AWS SCT

Sie können AWS SCT verwenden, um Schemas, Codeobjekte und Anwendungscode aus der Azure SQL-Datenbank in die folgenden Ziele zu konvertieren:

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition

Themen

- [Berechtigungen für Azure SQL Database als Quelle](#)
- [Verbindung zur Azure SQL-Datenbank als Quelle herstellen](#)

Berechtigungen für Azure SQL Database als Quelle

Die für Azure SQL Database als Quelle erforderlichen Rechte lauten wie folgt:

- VIEW DEFINITION
- VIEW DATABASE STATE

Gewähren Sie die erforderlichen Berechtigungen für alle Datenbanken, deren Schemata Sie konvertieren.

Die für MySQL- und PostgreSQL-Zieldatenbanken erforderlichen Rechte werden in den folgenden Abschnitten beschrieben.

- [Rechte für MySQL als Zieldatenbank](#)
- [Rechte für PostgreSQL als Zieldatenbank](#)

Verbindung zur Azure SQL-Datenbank als Quelle herstellen

Verwenden Sie das folgende Verfahren, um mit dem eine Verbindung zu Ihrer Azure SQL Database-Quelldatenbank herzustellenAWS Schema Conversion Tool.

So stellen Sie eine Verbindung zu einer Azure SQL Database-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Azure SQL Database und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen der Azure SQL-Datenbank-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Quelldatenbankservers ein.
Datenbank	Geben Sie den Datenbanknamen ein, zu dem eine Verbindung hergestellt werden soll.
User name und Password	<p>Geben Sie die Datenbankmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>

Parameter	Action
Passwort speichern	AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Verwendung von IBM Db2 for z/OS als Quelle für AWS SCT

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von IBM Db2 for z/OS in die folgenden Ziele zu konvertieren.

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition

Voraussetzungen für Db2 for z/OS als Quelldatenbank

Die Datenbankversion von IBM Db2 for z/OS Version 12 Function Level 100 unterstützt die meisten neuen Funktionen von IBM Db2 for z/OS Version 12 nicht. Diese Datenbankversion unterstützt den Fallback auf Db2 Version 11 und die gemeinsame Nutzung von Daten mit Db2 Version 11. Um die Konvertierung nicht unterstützter Funktionen von Db2 Version 11 zu vermeiden, empfehlen wir, dass Sie eine IBM Db2 for z/OS-Datenbank der Funktionsstufe 500 oder höher als Quelle für verwenden. AWS SCT

Sie können das folgende Codebeispiel verwenden, um die Version Ihrer IBM Db2 for z/OS-Quelldatenbank zu überprüfen.

```
SELECT GETVARIABLE('SYSIBM.VERSION') as version FROM SYSIBM.SYSDUMMY1;
```

Stellen Sie sicher, dass dieser Code Version DSN12015 oder höher zurückgibt.

Sie können das folgende Codebeispiel verwenden, um den Wert des APPLICATION COMPATIBILITY speziellen Registers in Ihrer IBM Db2 for z/OS-Quelldatenbank zu überprüfen.

```
SELECT CURRENT APPLICATION COMPATIBILITY as version FROM SYSIBM.SYSDUMMY1;
```

Stellen Sie sicher, dass dieser Code Version V12R1M500 oder höher zurückgibt.

Rechte für Db2 for z/OS als Quelldatenbank

Die erforderlichen Rechte, um eine Verbindung zu einer Db2 for z/OS-Datenbank herzustellen und Systemkataloge und Tabellen zu lesen, lauten wie folgt:

- AUF SYSIBM.LOCATIONS AUSWÄHLEN
- WÄHLEN SIE AUF SYSIBM.SYSCHECKS
- WÄHLE AUF SYSIBM.SYSCOLUMNS
- AUF SYSIBM.SYSDATABASE AUSWÄHLEN
- WÄHLEN SIE AUS SYSIBM.SYSDATATYPES
- WÄHLE AUF SYSIBM.SYSDUMMY1
- WÄHLE AUF SYSIBM.SYSFOREIGNKEYS
- WÄHLEN SIE AUF SYSIBM.SYSINDEXES
- WÄHLEN SIE AUF SYSIBM.SYSKEYCOLUSE
- WÄHLE AUF SYSIBM.SYSKEYS
- WÄHLE AUF SYSIBM.SYSKEYTARGETS
- WÄHLEN SIE AUF SYSIBM.SYSJAROBJECTS
- WÄHLEN SIE EIN SYSIBM.SYSPACKAGE
- WÄHLE AUF SYSIBM.SYSPARMS
- WÄHLEN SIE AUF SYSIBM.SYSRELS
- WÄHLEN SIE AUF SYSIBM.SYSROUTINES
- WÄHLE AUF SYSIBM.SYSSEQUENCES
- WÄHLEN SIE AUF SYSIBM.SYSSEQUENCESDEP
- WÄHLE SYSIBM.SYSSYNONYME
- WÄHLE AUF SYSIBM.SYSTABCONST

- WÄHLE AUF SYSIBM.SYSTABLES
- WÄHLE AUF SYSIBM.SYSTABLESPACE
- WÄHLE AUF SYSIBM.SYSTRIGGERS
- WÄHLE AUF SYSIBM.SYSVARIABLES
- WÄHLEN SIE AUF SYSIBM.SYSVIEWS

Um Db2 for z/OS-Tabellen in partitionierte PostgreSQL-Tabellen zu konvertieren, sammeln Sie mithilfe des Hilfsprogramms Statistiken über Tablespaces und Tabellen in Ihrer Datenbank, wie im Folgenden gezeigt. RUNSTATS

```
LISTDEF YOURLIST INCLUDE TABLESPACES DATABASE YOURDB
RUNSTATS TABLESPACE
LIST YOURLIST
TABLE (ALL) INDEX (ALL KEYCARD)
UPDATE ALL
REPORT YES
SHRLEVEL REFERENCE
```

Ersetzen Sie im vorherigen Beispiel den *YOURDB* Platzhalter durch den Namen der Quelldatenbank.

Verbindung zu Db2 für z/OS als Quelle herstellen

Gehen Sie wie folgt vor, um mit Ihrer Db2 for z/OS-Quelldatenbank eine Verbindung herzustellen.
AWS SCT

So stellen Sie eine Verbindung zu einer IBM Db2 for z/OS-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Db2 for z/OS und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:

1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen zur IBM Db2 for z/OS-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Ort	Geben Sie den eindeutigen Namen des Db2-Standorts ein, auf den Sie zugreifen möchten.
User name und Password	<p>Geben Sie die Datenbankmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>

Parameter	Action
Use SSL	<p>Wählen Sie diese Option, wenn Sie Secure Sockets Layer (SSL) verwenden möchten, um eine Verbindung zu Ihrer Datenbank herzustellen. Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an:</p> <ul style="list-style-type: none"> Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält. Damit dieser Standort hier angezeigt wird, stellen Sie sicher, dass Sie ihn in den globalen Einstellungen hinzufügen.
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankkennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.</p>
Treiberpfad für Db2 für z/OS	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projekteinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

- Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden kann.
- Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Rechte für MySQL als Zieldatenbank

Die für MySQL als Ziel erforderlichen Rechte lauten wie folgt:

- ERSTELLEN AM * . *

- ÄNDERN AM *.*
- VORBEIKOMMEN *.*
- INDEX AUF *.*
- VERWEISE AUF *.*
- SELECT ON *.*
- ANSICHT ERSTELLEN AM *.*
- SHOW VIEW ON *.*
- AUSLÖSER EIN*.*
- ROUTINE ERSTELLEN AM *.*
- ROUTINE ÄNDERN AM *.*
- AUSFÜHREN AM *.*
- SELECT ON mysql.proc
- EINFÜGEN, AUF AWS_DB2ZOS_EXT AKTUALISIEREN.*
- AUF AWS_DB2ZOS_EXT_DATA EINFÜGEN, AKTUALISIEREN, LÖSCHEN.*
- ERSTELLT TEMPORÄRE TABELLEN AUF AWS_DB2ZOS_EXT_DATA.*

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Rechte zu gewähren.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT SELECT ON mysql.proc TO 'user_name';
GRANT INSERT, UPDATE ON AWS_DB2ZOS_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_DB2ZOS_EXT_DATA.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_DB2ZOS_EXT_DATA.* TO 'user_name';
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen *Sie dann your_password* durch ein sicheres Passwort.

Um Amazon RDS for MySQL als Ziel zu verwenden, setzen Sie den `log_bin_trust_function_creators` Parameter auf `true` und den `character_set_server` auf `latin1`. Um diese Parameter zu konfigurieren, erstellen Sie eine neue DB-Parametergruppe oder ändern Sie eine bestehende DB-Parametergruppe.

Um Aurora MySQL als Ziel zu verwenden, setzen Sie den `log_bin_trust_function_creators` Parameter auf `true` und den Parameter `character_set_server` auf `latin1`. Stellen Sie den `lower_case_table_names` Parameter außerdem auf `True` ein. Um diese Parameter zu konfigurieren, erstellen Sie eine neue DB-Parametergruppe oder ändern Sie eine bestehende DB-Parametergruppe.

Privilegien für PostgreSQL als Zieldatenbank

Um PostgreSQL als Ziel zu verwenden, ist das AWS SCT Privileg erforderlich. `CREATE ON DATABASE` Stellen Sie sicher, dass Sie dieses Privileg für jede PostgreSQL-Zieldatenbank gewähren.

Um Amazon RDS for PostgreSQL als Ziel zu verwenden, ist das Privileg `AWS SCT` erforderlich.
`rds_superuser`

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in `"$user", public_synonyms, public`.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Rechte zu gewähren.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';  
GRANT CREATE ON DATABASE db_name TO user_name;  
GRANT rds_superuser TO user_name;  
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie dann *db_name* durch den Namen Ihrer Zieldatenbank. Ersetzen *Sie abschließend your_password* durch ein sicheres Passwort.

In PostgreSQL `superuser` kann nur der Schemabesitzer oder `a` ein Schema löschen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Besitzer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, erhalten Sie möglicherweise eine Fehlermeldung, wenn ein Schema nicht gelöscht AWS SCT werden kann. Verwenden Sie die `superuser` Rolle, um diese Fehlermeldung zu vermeiden.

Einstellungen für die Konvertierung von Db2 für z/OS in PostgreSQL

Um die Konvertierungseinstellungen von Db2 für z/OS in PostgreSQL zu bearbeiten, wählen Sie Einstellungen und dann Konvertierungseinstellungen. Wählen Sie aus der oberen Liste Db2 für z/OS und dann Db2 für z/OS — PostgreSQL oder Db2 für z/OS — Amazon Aurora (PostgreSQL-kompatibel). AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von IBM Db2 für z/OS in PostgreSQL an.

Die Einstellungen für die Konvertierung von Db2 für z/OS in PostgreSQL AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um eindeutige Namen für Einschränkungen in der Zieldatenbank zu generieren.

In PostgreSQL müssen alle Constraint-Namen, die Sie verwenden, eindeutig sein. AWS SCT kann eindeutige Namen für Beschränkungen im konvertierten Code generieren, indem Sie dem Namen Ihrer Einschränkung ein Präfix mit dem Tabellennamen hinzufügen. Um sicherzustellen, dass eindeutige Namen für Ihre Einschränkungen AWS SCT generiert werden, wählen Sie Eindeutige Namen für Einschränkungen generieren aus.

- Um die Formatierung von Spaltennamen, Ausdrücken und Klauseln in DML-Anweisungen im konvertierten Code beizubehalten.

AWS SCT kann das Layout von Spaltennamen, Ausdrücken und Klauseln in DML-Anweisungen an der ähnlichen Position und Reihenfolge wie im Quellcode beibehalten. Wählen Sie dazu Ja für Die Formatierung von Spaltennamen, Ausdrücken und Klauseln in DML-Anweisungen beibehalten aus.

- Um Tabellenpartitionen aus dem Konvertierungsumfang auszuschließen.

AWS SCT kann während der Konvertierung alle Partitionen einer Quelltablelle überspringen.

Wählen Sie dazu die Option Tabellenpartitionen aus dem Konvertierungsbereich ausschließen aus.

- Um die automatische Partitionierung für Tabellen zu verwenden, die nach Wachstum partitioniert sind.

AWS SCT kann für die Datenmigration automatisch alle Tabellen partitionieren, die größer als die angegebene Größe sind. Um diese Option zu verwenden, wählen Sie Partition für Tabellen erzwingen, die größer sind als, und geben Sie die Tabellengröße in Gigabyte ein. Geben Sie als Nächstes die Anzahl der Partitionen ein. AWS SCT berücksichtigt die Größe des Direct Access Storage Device (DASD) Ihrer Quelldatenbank, wenn Sie diese Option aktivieren.

AWS SCT kann die Anzahl der Partitionen automatisch ermitteln. Wählen Sie dazu die Option Anzahl der Partitionen proportional erhöhen und geben Sie die maximale Anzahl von Partitionen ein.

- Um dynamische Ergebnismengen als Array von Werten des Refcursor-Datentyps zurückzugeben.

AWS SCT kann Quellprozeduren, die dynamische Ergebnismengen zurückgeben, in Prozeduren konvertieren, die ein Array offener Refcursoren als zusätzlichen Ausgabeparameter haben. Wählen Sie dazu Use an array of refcursors to return all dynamic result sets aus.

- Um den Standard anzugeben, der für die Konvertierung von Datums- und Uhrzeitwerten in Zeichenkettendarstellungen verwendet werden soll.

AWS SCT kann Datums- und Uhrzeitwerte mithilfe eines der unterstützten Branchenformate in Zeichenkettendarstellungen konvertieren. Wählen Sie dazu Zeichenfolgendarstellungen von Datumswerten verwenden oder Zeichenfolgendarstellungen von Zeitwerten verwenden aus. Wählen Sie als Nächstes einen der folgenden Standards aus.

- Internationale Normenorganisation (ISO)
- IBM Europäischer Standard (EUR)
- IBM USA Standard (Vereinigte Staaten)
- Japanischer Industriestandard Christian Era (JIS)

Verwendung von IBM Db2 LUW als Quelle für AWS SCT

Sie können AWS SCT damit Schemas, Codeobjekte in der SQL-Sprache und Anwendungscode von IBM Db2 für Linux, Unix und Windows (Db2 LUW) in die folgenden Ziele konvertieren.

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon RDS for MariaDB

AWS SCT unterstützt als Quelle die Db2-LUW-Versionen 9.1, 9.5, 9.7, 10.1, 10.5, 11.1 und 11.5.

Rechte für Db2 LUW als Quelle

Die erforderlichen Rechte, um eine Verbindung zu einer Db2-LUW-Datenbank herzustellen, die verfügbaren Rechte zu überprüfen und Schemametadaten für eine Quelle zu lesen, lauten wie folgt:

- Berechtigung, die erforderlich sind, um eine Verbindung herzustellen:
 - VERBINDUNG ZUR DATENBANK HERSTELLEN
- Berechtigung, die zum Ausführen von SQL-Anweisungen erforderlich ist:
 - IM PAKET NULLID.SYSSH200 AUSFÜHREN
- Berechtigungen, die für den Abruf von Informationen auf Instance-Ebene erforderlich sind:
 - FÜHRE DIE FUNKTION SYSPROC.ENV_GET_INST_INFO AUS
 - WÄHLE AUF SYSIBMADM.ENV_INST_INFO
 - WÄHLE AUF SYSIBMADM.ENV_SYS_INFO
- Berechtigungen, die benötigt werden, um Berechtigungen zu prüfen, die durch Rollen, Gruppen und Behörden vergeben werden:
 - FÜHRE DIE FUNKTION SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID AUS
 - FÜHRE DIE FUNKTION SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID AUS
 - FÜHRE DIE FUNKTION SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID AUS
 - WÄHLEN SIE SYSIBMADM.PRIVILEGES
- Berechtigungen, die für System-Kataloge und Tabellen benötigt werden:
 - WÄHLE AUF SYSCAT.ATTRIBUTES
 - AUF SYSCAT.CHECKS AUSWÄHLEN
 - WÄHLEN SIE AUF SYSCAT.COLIDENTATTRIBUTES
 - AUF SYSCAT.COLUMNS AUSWÄHLEN

- WÄHLEN SIE AUF SYSCAT.DATAPARTITIONEXPRESSION AUS
- WÄHLEN SIE AUF SYSCAT.DATAPARTITIONS
- WÄHLE AUF SYSCAT.DATATYPEDEP
- WÄHLEN SIE SYSCAT.DATATYPES
- AUF SYSCAT.HIERARCHIES AUSWÄHLEN
- WÄHLEN SIE AUF SYSCAT.INDEXCLUSE
- AUF SYSCAT.INDEXES AUSWÄHLEN
- WÄHLEN SIE AUF SYSCAT.INDEXPARTITIONS
- WÄHLE AUF SYSCAT.KEYCLUSE
- WÄHLE AUF SYSCAT.MODULEOBJECTS
- WÄHLE AUF SYSCAT.MODULES
- WÄHLE AUF SYSCAT.NICKNAMES
- WÄHLE AUF SYSCAT.PERIODS
- AUF SYSCAT.REFERENCES AUSWÄHLEN
- WÄHLE AUF SYSCAT.ROUTINEPARMS
- WÄHLE AUF SYSCAT.ROUTINES
- WÄHLE AUF SYSCAT.ROWFIELDS
- WÄHLE AUF SYSCAT.SCHEMATA
- WÄHLE AUF SYSCAT.SEQUENCES
- WÄHLE AUF SYSCAT.TABCONST
- AUF SYSCAT.TABLES AUSWÄHLEN
- WÄHLE AUF SYSCAT.TRIGGERS
- WÄHLE AUF SYSCAT.VARIABLEDEP
- WÄHLE AUF SYSCAT.VARIABLES
- AUF SYSCAT.VIEWS AUSWÄHLEN
- WÄHLE AUF SYSIBM.SYSDUMMY1
- Für die Ausführung von SQL-Anweisungen benötigt das Benutzerkonto eine Berechtigung, mindestens eine der in der Datenbank aktivierten Arbeitslasten zu verwenden. Wurde dem Benutzer keine der Arbeitslasten zugeordnet, stellen Sie sicher, dass der Benutzer Zugriff auf die Standard-Benutzerarbeitslast hat:

- VERWENDUNG AUF WORKLOAD SYSDEFAULTUSERWORKLOAD

Um Abfragen auszuführen, müssen Sie temporäre System-Tabellenräume mit den Seitengrößen 8K, 16K und 32K anlegen, falls diese nicht existieren. Um die temporären Tabellenräume zu erstellen, führen Sie die folgenden Skripts aus:

```
CREATE BUFFERPOOL BP8K
  IMMEDIATE
  ALL DBPARTITIONNUMS
  SIZE AUTOMATIC
  NUMBLOCKPAGES 0
  PAGESIZE 8K;

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_8K
  PAGESIZE 8192
  BUFFERPOOL BP8K;

CREATE BUFFERPOOL BP16K
  IMMEDIATE
  ALL DBPARTITIONNUMS
  SIZE AUTOMATIC
  NUMBLOCKPAGES 0
  PAGESIZE 16K;

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_BP16K
  PAGESIZE 16384
  BUFFERPOOL BP16K;

CREATE BUFFERPOOL BP32K
  IMMEDIATE
  ALL DBPARTITIONNUMS
  SIZE AUTOMATIC
  NUMBLOCKPAGES 0
  PAGESIZE 32K;

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_BP32K
  PAGESIZE 32768
  BUFFERPOOL BP32K;
```

Verbindung zu Db2 LUW als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Db2 LUW-Quelldatenbank herzustellen.

Herstellen einer Verbindung zu einer Db2 LUW-Quelldatenbank

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Db2 LUW und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen der IBM Db2 LUW-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Db2 LUW-Datenbank ein.
User name und Password	Geben Sie die Datenbankmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie

Parameter	Action
	<p>sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>
Use SSL	<p>Wählen Sie diese Option, wenn Sie Secure Sockets Layer (SSL) verwenden möchten, um eine Verbindung zu Ihrer Datenbank herzustellen. Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an:</p> <ul style="list-style-type: none">• Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält. Damit dieser Standort hier angezeigt wird, stellen Sie sicher, dass Sie ihn in den globalen Einstellungen hinzufügen.
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankkennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.</p>
Db2 LUW-Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projektinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Konvertierung von Db2 LUW zu Amazon RDS for PostgreSQL oder Amazon Aurora PostgreSQL-Compatible Edition

Wenn Sie IBM Db2 LUW in PostgreSQL migrieren, kann AWS SCT verschiedene Auslöseranweisungen umwandeln, die mit Db2 LUW verwendet werden. Zu diesen Auslöseranweisungen gehören:

- Triggerereignisse — Die Triggerereignisse INSERT, DELETE und UPDATE geben an, dass die ausgelöste Aktion immer dann ausgeführt wird, wenn das Ereignis auf die Betrefftable oder die Betreffansicht angewendet wird. Sie können eine beliebige Kombination der INSERT-, DELETE- und UPDATE-Ereignisse angeben, jedes Ereignis kann aber nur einmal angegeben werden. AWS SCT unterstützt einzelne und mehrere Auslöserereignisse. Bei Ereignissen hat PostgreSQL praktisch die gleiche Funktionalität.
- Event OF COLUMN — Sie können einen Spaltennamen aus einer Basistabelle angeben. Der Auslöser wird nur durch die Aktualisierung einer Spalte ausgelöst, die in der Liste der Spaltennamen identifiziert wird. PostgreSQL hat dieselbe Funktionalität.
- Statement-Trigger — Diese geben an, dass die ausgelöste Aktion nur einmal für die gesamte Anweisung angewendet wird. Sie können diese Art von Auslösergranularität nicht für einen BEFORE-Auslöser oder einen INSTEAD OF-Auslöser anwenden. Sofern dieser Parameter angegeben ist, wird ein UPDATE- oder DELETE-Auslöser aktiviert. Dies gilt auch dann, wenn keine Zeilen betroffen sind. PostgreSQL verfügt auch über diese Funktionalität und die Auslöserdeklaration für Anweisungsauslöser ist für PostgreSQL und Db2 LUW identisch.
- Referenzklauseln — Diese spezifizieren die Korrelationsnamen für Übergangsvariablen und die Tabellennamen für Übergangstabellen. Korrelationsnamen identifizieren eine bestimmte Zeile in der Gruppe von Zeilen, die von der auslösenden SQL-Operation betroffen sind. Tabellennamen identifizieren die vollständige Menge der betroffenen Zeilen. Jede durch eine auslösende SQL-Operation betroffene Zeile ist für die ausgelöste Aktion anhand qualifizierender Spalten mit angegebenen Korrelationsnamen verfügbar. PostgreSQL unterstützt diese Funktionalität nicht und verwendet nur den Korrelationsnamen NEW oder OLD.
- ANSTELLE VON Triggern — AWS SCT unterstützt diese.

Konvertierung partitionierter Db2 LUW-Tabellen in partitionierte PostgreSQL Version 10-Tabellen

AWS SCT kann Db2 LUW-Tabellen zu partitionierten Tabellen in PostgreSQL Version 10 umwandeln. Es gibt mehrere Einschränkungen beim Umwandeln einer partitionierten Db2 LUW-Tabelle in PostgreSQL:

- Sie können in Db2 LUW eine partitionierte Tabelle mit einer löschbaren Spalte erstellen, und Sie können eine Partition angeben, die NULL-Werte speichert. PostgreSQL unterstützt jedoch keine NULL-Werte für RANGE-Partitionierungen.
- LUW Db2 kann in einer INCLUSIVE- oder EXCLUSIVE-Klausel Bereichsgrenzenwerte festlegen. PostgreSQL unterstützt nur INCLUSIVE für ein Startgrenze und EXCLUSIVE für eine Endgrenze. Der Name der konvertierten Partition hat das Format `<original_table_name>_<original_partition_name>`.
- Für partitionierte Tabellen in Db2 LUW können Sie primäre oder eindeutige Schlüssel erstellen. In PostgreSQL müssen Sie primäre oder eindeutige Schlüssel direkt für jede Partition erstellen. Primäre oder eindeutige Schlüsselbeschränkungen müssen aus der übergeordneten Tabelle entfernt werden. Der Name des konvertierten Schlüssels hat das Format `<original_key_name>_<original_partition_name>`.
- Sie können eine Fremdschlüsselbeschränkung von und zu einer partitionierten Tabelle in der Db2-LUW erstellen. PostgreSQL unterstützt jedoch keine Fremdschlüsselverweise in partitionierten Tabellen. PostgreSQL unterstützt auch keine Fremdschlüsselverweise aus einer partitionierten Tabelle in eine andere Tabelle.
- Sie können einen Index für eine partitionierte Tabelle in Db2 LUW erstellen. In PostgreSQL müssen Sie jedoch einen Index direkt für jede Partition erstellen. Indizes müssen aus der übergeordneten Tabelle entfernt werden. Der Name des konvertierten Index hat das Format `<original_index_name>_<original_partition_name>`.
- Sie müssen Zeilenauslöser für einzelne Partitionen definieren, nicht für die partitionierte Tabelle. Auslöser müssen aus der übergeordneten Tabelle entfernt werden. Der Name des konvertierten Auslösers hat das Format `<original_trigger_name>_<original_partition_name>`.

Privilegien für PostgreSQL als Ziel

Um PostgreSQL als Ziel zu verwenden, ist das AWS SCT Privileg erforderlich. `CREATE ON DATABASE` Stellen Sie sicher, dass Sie dieses Privileg für jede PostgreSQL-Zieldatenbank gewähren.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in "\$user", public_synonyms, public.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Rechte zu gewähren.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';  
GRANT CREATE ON DATABASE db_name TO user_name;  
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie dann *db_name* durch den Namen Ihrer Zieldatenbank. Ersetzen *Sie abschließend your_password* durch ein sicheres Passwort.

In PostgreSQL superuser kann nur der Schemabesitzer oder a ein Schema löschen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Besitzer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, erhalten Sie möglicherweise eine Fehlermeldung, wenn ein Schema nicht gelöscht AWS SCT werden kann. Verwenden Sie die superuser Rolle, um diese Fehlermeldung zu vermeiden.

Konvertierung von Db2 LUW zu Amazon RDS für MySQL oder Amazon Aurora MySQL

Beachten Sie Folgendes, wenn Sie eine IBM Db2 LUW-Datenbank in RDS für MySQL oder Amazon Aurora MySQL konvertieren.

Privilegien für MySQL als Ziel

Die für MySQL als Ziel erforderlichen Rechte lauten wie folgt:

- ERSTELLEN AM * . *
- ÄNDERN AM * . *
- VORBEIKOMMEN * . *
- INDEX AUF * . *
- VERWEISE AUF * . *
- SELECT ON *.*

- ANSICHT ERSTELLEN AM *.*
- SHOW VIEW ON *.*
- AUSLÖSER EIN*.*
- ROUTINE ERSTELLEN AM *.*
- ROUTINE ÄNDERN AM *.*
- AUSFÜHREN AM *.*
- SELECT ON mysql.proc
- EINFÜGEN, AUF AWS_DB2_EXT AKTUALISIEREN.*
- AUF AWS_DB2_EXT_DATA EINFÜGEN, AKTUALISIEREN, LÖSCHEN.*
- ERSTELLT TEMPORÄRE TABELLEN AUF AWS_DB2_EXT_DATA.*

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Rechte zu gewähren.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT SELECT ON mysql.proc TO 'user_name';
GRANT INSERT, UPDATE ON AWS_DB2_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_DB2_EXT_DATA.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_DB2_EXT_DATA.* TO 'user_name';
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen *Sie dann your_password* durch ein sicheres Passwort.

Um Amazon RDS for MySQL oder Aurora MySQL als Ziel zu verwenden, setzen Sie den `lower_case_table_names` Parameter auf 1. Dieser Wert bedeutet, dass der MySQL-Server

Bezeichner von Objektnamen wie Tabellen, Indizes, Triggern und Datenbanken so behandelt, dass die Groß- und Kleinschreibung nicht berücksichtigt wird. Wenn Sie die binäre Protokollierung in Ihrer Zielinstanz aktiviert haben, setzen Sie den `log_bin_trust_function_creators` Parameter auf 1. In diesem Fall müssen Sie die `READS SQL DATA` oder `NO SQL` Eigenschaften nicht verwenden `DETERMINISTIC`, um gespeicherte Funktionen zu erstellen. Um diese Parameter zu konfigurieren, erstellen Sie eine neue DB-Parametergruppe oder ändern Sie eine bestehende DB-Parametergruppe.

Verwendung von MySQL als Quelle für AWS SCT

Sie können AWS SCT verwenden, um Schemas, Datenbankcodeobjekte und Anwendungscode von MySQL in die folgenden Ziele zu konvertieren:

- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon RDS for MySQL

Weitere Informationen finden Sie in den folgenden Abschnitten:

Themen

- [Berechtigungen für MySQL als Quelldatenbank](#)
- [Verbindung zu MySQL als Quelle](#)
- [Privilegien für PostgreSQL als Zieldatenbank](#)

Berechtigungen für MySQL als Quelldatenbank

Die für MySQL als Quelle erforderlichen Rechte lauten wie folgt:

- `SELECT ON *.*`
- `SHOW VIEW ON *.*`

Verbindung zu MySQL als Quelle

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer MySQL-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer MySQL-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen.
2. Wählen Sie MySQL und dann Weiter.

Das Dialogfeld.

3. Geben Sie unter Verbindungs-Name für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankanmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um Datenbankanmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWS Secret den Namen des Secrets aus.
 2. Wählen Sie Füllen, um automatisch alle Werte im Dialogfeld für die Datenbankverbindung aus Secrets Manager einzugeben.

Hinweise zur Verwendung von Datenbankanmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen der MySQL-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	<p>Geben Sie den DNS) -----Quelldatenbankserver (DNS) -----</p> <p>Sie dafür ein IPv6-Adressprotokoll, wie ein IPv6-Adressprotokoll. Stellen Sie dafür sicher, dass Sie für die IP-Adresse sicher, dass Sie die IP-Adresse, wie im folgenden Beispiel gezeigt.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;">[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</div>
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.

Parameter	Action
User name und Password	<p data-bbox="656 226 1419 304">Geben Sie die Datenbank-Anmeldeinformationen zum Quelldatenbank-Server ein.</p> <p data-bbox="656 352 1500 766">AWS SCT verwendet das Passwort nur, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie in einem Projekt eine Verbindung zu Ihrer Datenbank herstellen möchten. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert es das Passwort standardmäßig nicht. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>
Use SSL	<p data-bbox="656 814 1495 1039">Wählen Sie diese Option, um Secure Sockets Layer (SSL) zum Verbinden Sie Ihre Datenbank (SSL) für die Verbindung mit Ihrer Datenbank. Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL ein:</p> <ul data-bbox="656 1060 1479 1144" style="list-style-type: none"><li data-bbox="656 1060 1479 1144">• SSL erforderlich: Wählen Sie diese Option, um nur über SSL eine Verbindung zum Server herzustellen. <p data-bbox="688 1186 1500 1512">Wenn Sie Require SSL aktivieren, ist eine Verbindung zum Server nur möglich, wenn der Server SSL unterstützt. Wenn Sie Require SSL nicht aktivieren und der Server SSL nicht unterstützt, können Sie sich ohne SSL mit dem Server verbinden. Weitere Informationen finden Sie unter Konfiguration von MySQL für die Verwendung sicherer Verbindungen.</p> <ul data-bbox="656 1533 1479 1764" style="list-style-type: none"><li data-bbox="656 1533 1479 1659">• Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Truststores zu überprüfen.<li data-bbox="656 1680 1479 1764">• Trust Store: Der Standort eines Truststores, der Zertifikate enthält.

Parameter	Action
Passwort speichern	AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankpasswort speichern und schnell auf die Datenbank zugreifen, ohne dass Sie das Passwort eingeben müssen.
MySql Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projektinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

- Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden kann.
- Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Privilegien für PostgreSQL als Zieldatenbank

Um PostgreSQL als Ziel zu verwenden, ist das `CREATE ON DATABASE` Privileg erforderlich. Stellen Sie sicher, dass Sie dieses Privileg für jede PostgreSQL-Zieldatenbank gewähren.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in `"$user", public_synonyms, public`.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';
GRANT CREATE ON DATABASE db_name TO user_name;
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie dann *db_name* durch den Namen Ihrer Zieldatenbank. Ersetzen *Sie abschließend your_password* durch ein sicheres Passwort.

In PostgreSQL `superuser` kann nur der Schemabesitzer oder ein Schemabesitzer ein Schema löschen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Besitzer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, können Sie eine Fehlermeldung erhalten, wenn ein Schema nicht gelöscht werden kann. Verwenden Sie die `superuser` Rolle, um diese Fehlermeldung zu vermeiden.

Verwendung von Oracle Database als Quelle für AWS SCT

Sie können AWS SCT damit Schemas, Datenbankcode-Objekte und Anwendungscode aus Oracle Database in die folgenden Ziele konvertieren:

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon RDS for Oracle
- Amazon RDS for MariaDB

Wenn die Quelle eine Oracle-Datenbank ist, können Kommentare beispielsweise in einer PostgreSQL-Datenbank in das entsprechende Format konvertiert werden. AWS SCT kann Kommentare zu Tabellen, Ansichten und Spalten konvertieren. Kommentare können Apostrophe enthalten. AWS SCT Verdoppelt die Apostrophe bei der Konvertierung von SQL-Anweisungen, genau wie bei Zeichenkettenliteralen.

Weitere Informationen finden Sie unter den folgenden Topics.

Themen

- [Privilegien für Oracle als Quelle](#)
- [Verbindung zu Oracle als Quelle herstellen](#)
- [Konvertierung von Oracle zu Amazon RDS für PostgreSQL oder Amazon Aurora PostgreSQL](#)

- [Konvertierung von Oracle zu Amazon RDS für MySQL oder Amazon Aurora MySQL](#)
- [Umwandeln von Oracle in Amazon RDS für Oracle](#)

Privilegien für Oracle als Quelle

Die für Oracle als Quelle erforderlichen Rechte lauten wie folgt:

- CONNECT
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY
- WÄHLE AUF SYS.ARGUMENT\$

Verbindung zu Oracle als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Oracle-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Oracle-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Oracle und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWS Secret den Namen des Secrets.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen zur Oracle-Quelldatenbank manuell einzugeben:

Parameter	Action
Typ	<p>Wählen Sie den Verbindungstyp für die Datenbank aus. Geben Sie abhängig vom gewählten Typ die folgenden zusätzlichen Informationen ein:</p> <ul style="list-style-type: none">• SID<ul style="list-style-type: none">• Servername: Der DNS-Name (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers.• Server port: Der Port, der für die Verbindung zum Quelldatenbankserver verwendet wird• Oracle SID: Die Oracle-System-ID (SID). Um die Oracle-SID abzurufen, senden Sie die folgende Abfrage an Ihre Oracle-Datenbank: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre>• Name des Dienstes<ul style="list-style-type: none">• Server name: Der DNS-Name oder die IP-Adresse Ihres Quelldatenbankservers<p>Sie können mithilfe eines IPv6-Adressprotokolls eine Verbindung zu Ihrer Oracle-Quelldatenbank herstellen. Stellen Sie dazu sicher, dass Sie eckige Klammern verwenden, um die IP-Adresse einzugeben, wie im folgenden Beispiel gezeigt.</p><div data-bbox="716 1398 1507 1478" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; text-align: center;">[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</div>• Server port: Der Port, der für die Verbindung zum Quelldatenbankserver verwendet wird• Dienstname: Der Name des Oracle-Dienstes, zu dem eine Verbindung hergestellt werden soll. <ul style="list-style-type: none">• TNS-Alias

Parameter	Action
	<ul style="list-style-type: none">• TNS file path: Der Pfad zu der Datei, die die Transparent Network Substrate (TNS)-Namensverbindungsinformationen enthält <p>Nachdem Sie die TNS-Datei ausgewählt haben, werden alle Oracle-Datenbankverbindungen aus der Datei zur TNS-Aliasliste AWS SCT hinzugefügt.</p> <p>Wählen Sie diese Option, um eine Verbindung zu Oracle Real Application Clusters (RAC) herzustellen.</p> <ul style="list-style-type: none">• TNS-Alias: Der TNS-Alias aus dieser Datei, der für die Verbindung mit der Quelldatenbank verwendet werden soll.• TNS Connect-ID• TNS-Verbindungs-ID: Die Kennung für die registrierten TNS-Verbindungsinformationen.

Parameter	Action
User name und Password	<p data-bbox="656 226 1479 352">Geben Sie die Datenbankmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p data-bbox="656 405 1507 911">Wenn Sie zum ersten Mal eine Verbindung zur Oracle-Datenbank herstellen, geben Sie den Pfad zur Oracle-Treiberdatei (ojdbc8.jar) ein. Sie können die Datei unter http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html herunterladen. Stellen Sie sicher, dass Sie sich auf der kostenlosen Oracle Technical Network-Website registrieren, um den Download abzuschließen. AWS SCT verwendet den ausgewählten Treiber für alle zukünftigen Oracle-Datenbankverbindungen. Der Treiberpfad kann über die Registerkarte Treiber in den globalen Einstellungen geändert werden.</p> <p data-bbox="656 961 1507 1373">AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>

Parameter	Action
Use SSL	<p>Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden . Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an:</p> <ul style="list-style-type: none">• SSL-Authentifizierung: Wählen Sie diese Option, um die SSL-Authentifizierung per Zertifikat zu verwenden . Richten Sie Ihren Trust Store und Key Store unter Einstellungen, Allgemeine Einstellungen, Sicherheit ein.• Trust Store: Der zu verwendende Trust Store.• Schlüsselspeicher: Der zu verwendende Schlüssel speicher.
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wählen Sie diese Option, um das Datenbankkennwort zu speichern und schnell eine Verbindung zur Datenbank herzustellen, ohne das Passwort eingeben zu müssen.</p>
Oracle-Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projekteinstellungen speichern, wird der Treiberpfad nicht im Verbindungsdiaologfeld angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Konvertierung von Oracle zu Amazon RDS für PostgreSQL oder Amazon Aurora PostgreSQL

Beachten Sie Folgendes, wenn Sie eine Oracle-Datenbank in RDS for PostgreSQL oder Amazon Aurora PostgreSQL konvertieren.

Themen

- [Privilegien für PostgreSQL als Zieldatenbank](#)
- [Einstellungen für die Konvertierung von Oracle nach PostgreSQL](#)
- [Oracle-Sequenzen konvertieren](#)
- [Konvertieren von Oracle ROWID](#)
- [Dynamisches Oracle-SQL konvertieren](#)
- [Oracle-Partitionen konvertieren](#)

AWS SCT führt bei der Konvertierung von Oracle-Systemobjekten in PostgreSQL Konvertierungen durch, wie in der folgenden Tabelle dargestellt.

Oracle-Systemobjekt	Beschreibung	Konvertiertes PostgreSQL-Objekt
V\$VERSION	Zeigt die Versionsnummern der Core-Bibliothekskomponenten in der Oracle-Datenbank an	aws_oracle_ext.v\$version
V\$INSTANCE	Eine Ansicht, die den Status der aktuellen Instance anzeigt.	aws_oracle_ext.v\$instance

Sie können AWS SCT Oracle SQL*Plus-Dateien in psql konvertieren. Dabei handelt es sich um ein terminalbasiertes Frontend für PostgreSQL. Weitere Informationen finden Sie unter [Anwendungs-SQL konvertieren mit AWS SCT](#).

Privilegien für PostgreSQL als Zieldatenbank

Um PostgreSQL als Ziel zu verwenden, ist das AWS SCT Privileg erforderlich. CREATE ON DATABASE Stellen Sie sicher, dass Sie dieses Privileg für jede PostgreSQL-Zieldatenbank gewähren.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in "\$user", public_synonyms, public.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Rechte zu gewähren.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';  
GRANT CREATE ON DATABASE db_name TO user_name;  
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie dann *db_name* durch den Namen Ihrer Zieldatenbank. Ersetzen *Sie abschließend your_password* durch ein sicheres Passwort.

Um Amazon RDS for PostgreSQL als Ziel zu verwenden, ist das Privileg AWS SCT erforderlich.
rds_superuser

In PostgreSQL superuser kann nur der Schemabesitzer oder a ein Schema löschen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Besitzer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, erhalten Sie möglicherweise eine Fehlermeldung, wenn ein Schema nicht gelöscht AWS SCT werden kann. Verwenden Sie die superuser Rolle, um diese Fehlermeldung zu vermeiden.

Einstellungen für die Konvertierung von Oracle nach PostgreSQL

Um die Konvertierungseinstellungen von Oracle nach PostgreSQL zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Oracle und dann Oracle — PostgreSQL aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Oracle in PostgreSQL an.

Die Konvertierungseinstellungen von Oracle nach PostgreSQL AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die Konvertierung materialisierter Oracle-Ansichten in Tabellen oder materialisierte Ansichten in PostgreSQL zu ermöglichen AWS SCT. Wählen Sie für Materialized View-Konvertierung als aus, wie Ihre Materialized Views konvertiert werden sollen.
- Um mit Ihrem Oracle-Quellcode zu arbeiten, wenn er die TO_NUMBER Funktionen TO_CHAR TO_DATE, und mit Parametern enthält, die PostgreSQL nicht unterstützt. AWS SCT emuliert standardmäßig die Verwendung dieser Parameter im konvertierten Code.

Wenn Ihr Oracle-Quellcode nur Parameter enthält, die PostgreSQL unterstützt, können Sie native TO_CHAR PostgreSQL-Funktionen und -Funktionen verwenden. TO_DATE TO_NUMBER In diesem Fall funktioniert der konvertierte Code schneller. Um nur diese Parameter einzubeziehen, wählen Sie die folgenden Werte aus:

- Die Funktion TO_CHAR () verwendet keine Oracle-spezifischen Formatierungszeichenketten
- Die Funktion TO_DATE () verwendet keine Oracle-spezifischen Formatierungszeichenfolgen
- Die Funktion TO_NUMBER () verwendet keine Oracle-spezifischen Formatierungszeichenketten
- Um zu verhindern, dass Ihre Oracle-Quelldatenbank nur Ganzzahlwerte in den Primär- oder Fremdschlüsselspalten des NUMBER Datentyps speichert, AWS SCT können Sie diese Spalten in den BIGINT Datentyp konvertieren. Dieser Ansatz verbessert die Leistung Ihres konvertierten Codes. Um diesen Ansatz zu verfolgen, wählen Sie NUMBER-Primär- und Fremdschlüsselspalten in BIGINT-Spalten konvertieren aus. Stellen Sie sicher, dass Ihre Quelle keine Fließkommawerte in diesen Spalten enthält, um Datenverlust zu vermeiden.
- Um deaktivierte Trigger und Einschränkungen in Ihrem Quellcode zu überspringen. Wählen Sie dazu die Option Deaktivierte Trigger und Beschränkungen ignorieren.
- AWS SCT Zur Konvertierung von Zeichenfolgenvariablen, die als dynamisches SQL bezeichnet werden. Ihr Datenbankcode kann die Werte dieser Zeichenfolgenvariablen ändern. Um sicherzustellen, dass AWS SCT immer der neueste Wert dieser Zeichenfolgenvariablen konvertiert wird, wählen Sie Den dynamischen SQL-Code konvertieren, der in aufgerufenen Routinen erstellt wird.
- Um das zu beheben, unterstützen PostgreSQL Version 10 und frühere Versionen keine Prozeduren. Wenn Sie oder Ihre Benutzer mit der Verwendung von Prozeduren in PostgreSQL nicht vertraut sind, AWS SCT können Sie Oracle-Prozeduren in PostgreSQL-Funktionen konvertieren. Wählen Sie dazu die Option Prozeduren in Funktionen konvertieren aus.

- Um zusätzliche Informationen zu den aufgetretenen Aktionspunkten zu sehen. Dazu können Sie dem Erweiterungspaket bestimmte Funktionen hinzufügen, indem Sie Add on Exception Raise Block für Migrationsprobleme mit den nächsten Schweregraden auswählen. Wählen Sie dann die Schweregrade aus, um benutzerdefinierte Ausnahmen auszulösen.
- Um mit einer Oracle-Quelldatenbank zu arbeiten, die Einschränkungen für die automatisch generierten Namen enthalten kann. Wenn Ihr Quellcode diese Namen verwendet, stellen Sie sicher, dass Sie die Option Die vom System generierten Beschränkungsnamen mithilfe der ursprünglichen Quellnamen konvertieren auswählen. Wenn Ihr Quellcode diese Einschränkungen verwendet, aber nicht deren Namen verwendet, deaktivieren Sie diese Option, um die Konvertierungsgeschwindigkeit zu erhöhen.
- Um herauszufinden, ob Ihre Datenbank und Anwendungen in verschiedenen Zeitzonen ausgeführt werden. AWS SCT emuliert standardmäßig Zeitzonen im konvertierten Code. Sie benötigen diese Emulation jedoch nicht, wenn Ihre Datenbank und Anwendungen dieselbe Zeitzone verwenden. In diesem Fall entspricht die ausgewählte Zeitzone auf der Clientseite der Zeitzone auf dem Server.
- Um herauszufinden, ob Ihre Quell- und Zieldatenbanken in unterschiedlichen Zeitzonen laufen. Wenn dies der Fall ist, gibt die Funktion, die die SYSDATE integrierte Oracle-Funktion emuliert, im Vergleich zur Quellfunktion andere Werte zurück. Um sicherzustellen, dass Ihre Quell- und Zielfunktionen dieselben Werte zurückgeben, wählen Sie Set default time zone for SYSDATE-Emulation.
- Um die Funktionen der Oracle-Erweiterung in Ihrem konvertierten Code zu verwenden. Wählen Sie dazu unter Oracle-Implementierung verwenden die zu verwendenden Funktionen aus. Weitere Informationen zu Oracle finden Sie unter [Oracle](#) on GitHub

Oracle-Sequenzen konvertieren

AWS SCT konvertiert Sequenzen von Oracle nach PostgreSQL. Wenn Sie Sequenzen verwenden, um Integritätseinschränkungen aufrechtzuerhalten, stellen Sie sicher, dass sich die neuen Werte einer migrierten Sequenz nicht mit den vorhandenen Werten überschneiden.

Um konvertierte Sequenzen mit dem letzten Wert aus der Quelldatenbank zu füllen

1. Öffnen Sie Ihr AWS SCT Projekt mit Oracle als Quellcode.
2. Wählen Sie Einstellungen und dann Konvertierungseinstellungen.
3. Wählen Sie in der oberen Liste Oracle und dann Oracle — PostgreSQL aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Oracle in PostgreSQL an.

4. Wählen Sie Konvertierte Sequenzen mit dem letzten auf der Quellseite generierten Wert auffüllen.
5. Wählen Sie OK, um die Einstellungen zu speichern und das Dialogfeld mit den Konvertierungseinstellungen zu schließen.

Konvertieren von Oracle ROWID

In einer Oracle-Datenbank enthält die ROWID-Pseudospalte die Adresse der Tabellenzeile. Die ROWID-Pseudospalte ist einzigartig für Oracle, daher AWS SCT konvertiert sie die ROWID-Pseudospalte in eine Datenspalte in PostgreSQL. Durch diese Konvertierung können Sie die ROWID-Informationen behalten.

AWS SCT kann bei der Konvertierung der ROWID-Pseudospalte eine Datenspalte mit dem Datentyp erstellen. `bigint` Wenn kein Primärschlüssel vorhanden ist, wird die ROWID-Spalte als Primärschlüssel AWS SCT festgelegt. Wenn ein Primärschlüssel vorhanden ist, AWS SCT wird der ROWID-Spalte eine eindeutige Einschränkung zugewiesen.

Wenn Ihr Quelldatenbankcode Operationen mit ROWID enthält, die Sie nicht mit einem numerischen Datentyp ausführen AWS SCT können, können Sie eine Datenspalte mit dem `character varying` Datentyp erstellen.

So erstellen Sie eine Datenspalte für Oracle ROWID für ein Projekt

1. Öffnen Sie Ihr AWS SCT Projekt mit Oracle als Quellcode.
2. Wählen Sie Einstellungen und dann Konvertierungseinstellungen.
3. Wählen Sie in der oberen Liste Oracle und dann Oracle — PostgreSQL aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Oracle in PostgreSQL an.
4. Führen Sie für Generate row ID einen der folgenden Schritte aus:
 - Wählen Sie Als Identität generieren, um eine numerische Datenspalte zu erstellen.
 - Wählen Sie Als Charakterdomänentyp generieren, um eine Charakterdatenspalte zu erstellen.
5. Wählen Sie OK, um die Einstellungen zu speichern und das Dialogfeld mit den Konvertierungseinstellungen zu schließen.

Dynamisches Oracle-SQL konvertieren

Oracle bietet zwei Möglichkeiten, dynamisches SQL zu implementieren: mithilfe einer EXECUTE IMMEDIATE-Anweisung oder durch Aufrufen von Prozeduren im Paket DBMS_SQL. Wenn Ihre Oracle-Quelldatenbank Objekte mit dynamischem SQL enthält, verwenden Sie diese, AWS SCT um dynamische SQL-Anweisungen von Oracle in PostgreSQL zu konvertieren.

Um Oracle Dynamic SQL nach PostgreSQL zu konvertieren

1. Öffnen Sie Ihr AWS SCT Projekt mit Oracle als Quellcode.
2. Wählen Sie in der Oracle-Quellbaumansicht ein Datenbankobjekt aus, das dynamisches SQL verwendet.
3. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt, wählen Sie Schema konvertieren und stimmen Sie zu, dass die Objekte ersetzt werden, falls sie vorhanden sind. Der folgende Screenshot zeigt die konvertierte Prozedur unterhalb der Oracle-Prozedur mit dynamischem SQL.

The screenshot displays the AWS SCT interface with two panels. The left panel shows the Oracle source tree with the package PKG_DYNAMIC_CURSOR and its procedures. The right panel shows the Oracle procedure PROC_DYNAMIC_CURSOR1 and its converted PostgreSQL equivalent.

```
Oracle procedure: PROC_DYNAMIC_CURSOR1
01 PROCEDURE proc_dynamic_cursor1( v_docid INTEGER DEFAULT 111)
02 IS
03     TYPE cur_typ IS REF CURSOR;
04     cursor_o cur_typ;
05     SQL_str  VARCHAR2(1000);
06     Test_str VARCHAR2(1);
07 BEGIN
08     SQL_str := 'SELECT ''X'' || ' FROM TEST_ORA_PG.ACCOUNT' ||
09             ' WHERE id = :id_in';
10
11     OPEN cursor_o FOR SQL_str USING v_docid;
12     FETCH cursor_o INTO Test_str;
13     CLOSE cursor_o;
14 END
```

```
Amazon RDS for PostgreSQL function: pkg_dynamic_cursor$proc_dynamic_cursor1
01 CREATE OR REPLACE FUNCTION test_ora_pg.pkg_dynamic_cursor$proc_dynamic_cursor1(IN v_docid NUMERIC DEFAULT 111)
02 RETURNS void
03 AS
04 $BODY$
05 DECLARE
06     cursor_o REFCURSOR;
07     SQL_str CHARACTER VARYING(1000);
08     Test_str CHARACTER VARYING(1);
09 BEGIN
10     SQL_str := CONCAT_WS(' ', 'SELECT ' || 'X' || ' FROM ', 'test_ora_pg.account', 'WHERE id = $1');
11     OPEN cursor_o FOR
12     EXECUTE SQL_str USING v_docid;
13     FETCH cursor_o INTO Test_str;
14     CLOSE cursor_o;
15 END;
16 $BODY$
17 LANGUAGE plpgsql;
```

Oracle-Partitionen konvertieren

AWS SCT unterstützt derzeit die folgenden Partitionierungsmethoden:

- Bereich
- Auflisten
- Mehrspaltiger Bereich
- Hash
- Composite (Listenliste, Bereichsliste, Listenbereich, List-Hash, Bereichs-Hash, Hash-Hash)

Konvertierung von Oracle zu Amazon RDS für MySQL oder Amazon Aurora MySQL

Verwenden Sie das Oracle to MySQL-Erweiterungspaket, um Oracle-Datenbankfunktionen in Ihrem konvertierten MySQL-Code zu emulieren. AWS SCT Weitere Informationen zu Erweiterungspaketen finden Sie unter [Verwenden von AWS SCT Erweiterungspaketen](#).

Themen

- [Privilegien für MySQL als Zieldatenbank](#)
- [Einstellungen für die Konvertierung von Oracle nach MySQL](#)
- [Überlegungen zur Migration](#)
- [Konvertierung der WITH-Anweisung in Oracle nach RDS für MySQL oder Amazon Aurora MySQL](#)

Privilegien für MySQL als Zieldatenbank

Die für MySQL als Ziel erforderlichen Rechte lauten wie folgt:

- ERSTELLEN AM * . *
- ÄNDERN AM * . *
- VORBEIFAHREN* . *
- INDEX AUF * . *
- VERWEISE AUF * . *
- SELECT ON *.*
- ANSICHT ERSTELLEN AM * . *

- SHOW VIEW ON *.*
- AUSLÖSER EIN*.*
- ROUTINE ERSTELLEN AM *.*
- ROUTINE ÄNDERN AM *.*
- AUSFÜHREN AM *.*
- TEMPORÄRE TABELLEN ERSTELLEN AUF *.*
- AWS_LAMBDA_ACCESS
- EINFÜGEN, AUF AWS_ORACLE_EXT AKTUALISIEREN.*
- AUF AWS_ORACLE_EXT_DATA EINFÜGEN, AKTUALISIEREN, LÖSCHEN.*

Wenn Sie eine MySQL-Datenbankversion 5.7 oder niedriger als Ziel verwenden, gewähren Sie die INVOKE LAMBDA *.*-Berechtigung anstelle von AWS_LAMBDA_ACCESS. Erteilen Sie für MySQL-Datenbanken Version 8.0 und höher die AWS_LAMBDA_ACCESS-Berechtigung.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Rechte zu gewähren.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON *.* TO 'user_name';
GRANT AWS_LAMBDA_ACCESS TO 'user_name';
GRANT INSERT, UPDATE ON AWS_ORACLE_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_ORACLE_EXT_DATA.* TO 'user_name';
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen *Sie dann your_password* durch ein sicheres Passwort.

Wenn Sie eine MySQL-Datenbank Version 5.7 oder niedriger als Ziel verwenden, verwenden Sie `GRANT INVOKE LAMBDA ON *.* TO 'user_name'` stattdessen `GRANT AWS_LAMBDA_ACCESS TO 'user_name'`.

Um Amazon RDS for MySQL oder Aurora MySQL als Ziel zu verwenden, setzen Sie den `lower_case_table_names` Parameter auf 1. Dieser Wert bedeutet, dass der MySQL-Server Bezeichner von Objektnamen wie Tabellen, Indizes, Triggern und Datenbanken so behandelt, dass die Groß- und Kleinschreibung nicht berücksichtigt wird. Wenn Sie die binäre Protokollierung in Ihrer Zielinstanz aktiviert haben, setzen Sie den `log_bin_trust_function_creators` Parameter auf 1. In diesem Fall müssen Sie die `READS SQL DATA` oder `NO SQL` Eigenschaften nicht verwenden `DETERMINISTIC`, um gespeicherte Funktionen zu erstellen. Um diese Parameter zu konfigurieren, erstellen Sie eine neue DB-Parametergruppe oder ändern Sie eine bestehende DB-Parametergruppe.

Einstellungen für die Konvertierung von Oracle nach MySQL

Um die Konvertierungseinstellungen von Oracle nach MySQL zu bearbeiten AWS SCT, wählen Sie Einstellungen in und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Oracle und dann Oracle — MySQL aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Oracle in MySQL an.

Die Konvertierungseinstellungen von Oracle nach MySQL AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um dem entgegenzuwirken, kann Ihre Oracle-Quelldatenbank die ROWID Pseudospalte verwenden, aber MySQL unterstützt keine ähnliche Funktionalität. AWS SCT kann die ROWID Pseudospalte im konvertierten Code emulieren. Wählen Sie dazu Generate as identity für Generate row ID? .

Wenn Ihr Oracle-Quellcode die ROWID Pseudospalte nicht verwendet, wählen Sie Don't generate für Generate row ID? In diesem Fall funktioniert der konvertierte Code schneller.

- Um mit Ihrem Oracle-Quellcode zu arbeiten, wenn er die TO_CHAR TO_DATE, und TO_NUMBER Funktionen mit Parametern enthält, die MySQL nicht unterstützt. AWS SCT Emuliert standardmäßig die Verwendung dieser Parameter im konvertierten Code.

Wenn Ihr Oracle-Quellcode nur Parameter enthält, die PostgreSQL unterstützt, können Sie native MySQL- und TO_CHAR TO_NUMBER -Funktionen TO_DATE verwenden. In diesem Fall funktioniert der konvertierte Code schneller. Um nur diese Parameter einzubeziehen, wählen Sie die folgenden Werte aus:

- Die Funktion TO_CHAR () verwendet keine Oracle-spezifischen Formatierungszeichenketten
- Die Funktion TO_DATE () verwendet keine Oracle-spezifischen Formatierungszeichenfolgen
- Die Funktion TO_NUMBER () verwendet keine Oracle-spezifischen Formatierungszeichenketten
- Um festzustellen, ob Ihre Datenbank und Anwendungen in verschiedenen Zeitzonen laufen. AWS SCT Emuliert standardmäßig Zeitzonen im konvertierten Code. Sie benötigen diese Emulation jedoch nicht, wenn Ihre Datenbank und Anwendungen dieselbe Zeitzone verwenden. In diesem Fall entspricht die ausgewählte Zeitzone auf der Clientseite der Zeitzone auf dem Server.

Überlegungen zur Migration

Wenn Sie Oracle zu RDS für MySQL oder Aurora MySQL konvertieren, können Sie eine GOTO Anweisung und ein Label verwenden, um die Reihenfolge zu ändern, in der Anweisungen ausgeführt werden. Alle PL/SQL-Anweisungen, die auf eine GOTO Anweisung folgen, werden übersprungen und die Verarbeitung wird am Label fortgesetzt. Sie können GOTO Anweisungen und Labels überall innerhalb einer Prozedur, eines Batches oder eines Anweisungsblocks verwenden. Sie können auch als Nächstes GOTO-Statements aufrufen.

MySQL verwendet keine GOTO Anweisungen. Bei der AWS SCT Konvertierung von Code, der eine GOTO Anweisung enthält, wird die Anweisung konvertiert, sodass sie eine BEGIN...END LOOP...END LOOP ODER-Anweisung verwendet.

In der folgenden Tabelle finden Sie Beispiele für die AWS SCT Konvertierung von GOTO Anweisungen.

Oracle-Anweisung

```
BEGIN
  ....
  statement1;
  ....
  GOTO label1;
  statement2;
  ....
  label1:
  Statement3;
  ....
END
```

MySQL-Anweisung

```
BEGIN
  label1:
  BEGIN
    ....
    statement1;
    ....
    LEAVE label1;
    statement2;
    ....
  END;
  Statement3;
  ....
END
```

```
BEGIN
  ....
  statement1;
  ....
  label1:
  statement2;
  ....
  GOTO label1;
  statement3;
  ....
  statement4;
  ....
END
```

```
BEGIN
  ....
  statement1;
  ....
  label1:
  LOOP
    statement2;
    ....
    ITERATE label1;
    LEAVE label1;
  END LOOP;
  statement3;
  ....
  statement4;
  ....
END
```

```
BEGIN
  ....
  statement1;
  ....
  label1:
  statement2;
```

```
BEGIN
  ....
  statement1;
  ....
  label1:
  BEGIN
```

Oracle-Anweisung	MySQL-Anweisung
<pre> statement3; statement4; END </pre>	<pre> statement2; statement3; statement4; END; END </pre>

Konvertierung der WITH-Anweisung in Oracle nach RDS für MySQL oder Amazon Aurora MySQL

Sie verwenden die WITH-Klausel (subquery_factoring) in Oracle, um einem Unterabfragenblock einen Namen (query_name) zuzuweisen. Sie können dann an mehreren Stellen in der Abfrage auf den Unterabfragenblock Bezug nehmen. Wenn ein Unterabfrageblock keine Links oder Parameter (local, procedure, function, package) enthält, wird die Klausel in eine Ansicht oder eine temporäre Tabelle AWS SCT konvertiert.

Der Vorteil der Umwandlung der Klausel in eine temporäre Tabelle besteht darin, dass wiederholte Verweise auf die Unterabfrage effizienter sein können. Die größere Effizienz liegt darin, dass die Daten leicht aus der temporären Tabelle abgerufen werden können, anstatt von jeder Referenz benötigt zu werden. Sie können dies emulieren, indem Sie zusätzliche Ansichten oder eine temporäre Tabelle verwenden. Für den Ansichtsnamen wird das Format <procedure_name> \$<subselect_alias> verwendet.

Beispiele finden Sie in der folgenden Tabelle.

Oracle-Anweisung	MySQL-Anweisung
<pre> CREATE PROCEDURE TEST_ORA_PG.P_WITH_SELECT_V ARIABLE_01 (p_state IN NUMBER) AS l_dept_id NUMBER := 1; BEGIN FOR cur IN </pre>	<pre> CREATE PROCEDURE test_ora_pg.P_WITH _SELECT_VARIABLE_01(IN par_P_STATE DOUBLE) BEGIN DECLARE var_l_dept_id DOUBLE DEFAULT 1; DECLARE var\$id VARCHAR (8000); DECLARE var\$state VARCHAR (8000); </pre>

Oracle-Anweisung

```

        (WITH dept_empl(id, name,
surname,
        lastname, state, dept_id)
AS
        (
        SELECT id, name,
surname,
        lastname, state,
dept_id
        FROM test_ora_
pg.dept_employees
        WHERE state =
p_state AND
        dept_id =
l_dept_id)
SELECT id,state
FROM dept_empl
ORDER BY id) LOOP
NULL;
END LOOP;

```

MySQL-Anweisung

```

DECLARE done INT DEFAULT FALSE;
DECLARE cur CURSOR FOR SELECT
        ID, STATE
        FROM (SELECT
                ID, NAME, SURNAME,
LASTNAME, STATE, DEPT_ID
        FROM TEST_ORA_PG.DEPT_E
MPLOYEES
        WHERE STATE = par_p_sta
te AND DEPT_ID = var_l_dept_id) AS
dept_empl
        ORDER BY ID;
DECLARE CONTINUE HANDLER FOR NOT
FOUND
        SET done := TRUE;
OPEN cur;

read_label:
LOOP
        FETCH cur INTO var$id, var
$state;

        IF done THEN
                LEAVE read_label;
        END IF;

        BEGIN
        END;
END LOOP;
CLOSE cur;
END;

```

Oracle-Anweisung

```

CREATE PROCEDURE
  TEST_ORA_PG.P_WITH_SELECT_R
  EGULAR_MULT_01
AS
BEGIN

  FOR cur IN (
    WITH dept_emp1 AS
      (
        SELECT id,
name, surname,
          lastname,
state, dept_id
          FROM
test_ora_pg.dept_employees
          WHERE state =
1),
      dept AS
      (SELECT id deptid,
parent_id,
          name deptname
FROM test_ora_
pg.department
      )
    SELECT dept_emp1
.*,dept.*
          FROM dept_emp1, dept
          WHERE dept_emp1
.dept_id = dept.deptid
      ) LOOP
    NULL;
  END LOOP;

```

MySQL-Anweisung

```

CREATE VIEW TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept_emp1
` (id, name, surname, lastname, state,
dept_id)
AS
(SELECT id, name, surname, lastname,
state, dept_id
  FROM test_ora_pg.dept_employees
  WHERE state = 1);

CREATE VIEW TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept
` (deptid, parent_id,deptname)
AS
(SELECT id deptid, parent_id, name
deptname
  FROM test_ora_pg.department);

CREATE PROCEDURE test_ora_pg.P_WITH
_SELECT_REGULAR_MULT_01()
BEGIN
  DECLARE var$ID DOUBLE;
  DECLARE var$NAME VARCHAR (30);
  DECLARE var$SURNAME VARCHAR (30);
  DECLARE var$LASTNAME VARCHAR (30);
  DECLARE var$STATE DOUBLE;
  DECLARE var$DEPT_ID DOUBLE;
  DECLARE var$deptid DOUBLE;
  DECLARE var$PARENT_ID DOUBLE;
  DECLARE var$deptname VARCHAR
(200);
  DECLARE done INT DEFAULT FALSE;
  DECLARE cur CURSOR FOR SELECT
    dept_emp1.*, dept.*
    FROM TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept_emp1
    ` AS dept_emp1,
    TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept
    ` AS dept

```

Oracle-Anweisung	MySQL-Anweisung
	<pre>WHERE dept_empl.DEPT_ID = dept.DEPTID; DECLARE CONTINUE HANDLER FOR NOT FOUND SET done := TRUE; OPEN cur; read_label: LOOP FETCH cur INTO var\$ID, var\$NAME, var\$SURNAME, var\$LASTNAME, var\$STATE, var \$DEPT_ID, var\$deptid, var\$PARENT_ID, var\$deptname; IF done THEN LEAVE read_label; END IF; BEGIN END; END LOOP; CLOSE cur; END; call test_ora_pg.P_WITH_SELECT_R EGULAR_MULT_01()</pre>

Oracle-Anweisung

```

CREATE PROCEDURE
  TEST_ORA_PG.P_WITH_SELECT_V
  AR_CROSS_02(p_state IN NUMBER)
AS
  l_dept_id NUMBER := 10;
BEGIN
  FOR cur IN (
    WITH emp AS
      (SELECT id, name,
        surname,
          lastname, state,
            dept_id
          FROM test_ora_
pg.dept_employees
        WHERE dept_id >
  10
      ),
      active_emp AS
      (
        SELECT id
          FROM emp
        WHERE emp.state
= p_state
      )
    SELECT *
      FROM active_emp
    ) LOOP
    NULL;
  END LOOP;
END;

```

MySQL-Anweisung

```

CREATE VIEW TEST_ORA_PG.`P_WIT
H_SELECT_VAR_CROSS_01$emp
  `(id, name, surname, lastname,
    state, dept_id)
AS
(SELECT
  id, name, surname, lastname,
    state, dept_id
  FROM TEST_ORA_PG.DEPT_EMPLOYEES
  WHERE DEPT_ID > 10);

CREATE PROCEDURE
  test_ora_pg.P_WITH_SELECT_V
  AR_CROSS_02(IN par_P_STATE DOUBLE)
BEGIN
  DECLARE var_l_dept_id DOUBLE
  DEFAULT 10;
  DECLARE var$ID DOUBLE;
  DECLARE done INT DEFAULT FALSE;
  DECLARE cur CURSOR FOR SELECT *
    FROM
  (SELECT
    ID
  FROM
    TEST_ORA_
PG.
    `P_WITH_S
    ELECT_VAR_CROSS_01$emp` AS emp
  WHERE emp.STATE = par_p_state)
  AS
  active_emp;
  DECLARE CONTINUE HANDLER FOR NOT
  FOUND
    SET done := TRUE;
  OPEN cur;

  read_label:

```

Oracle-Anweisung	MySQL-Anweisung
	<pre>LOOP FETCH cur INTO var\$ID; IF done THEN LEAVE read_label; END IF; BEGIN END; END LOOP; CLOSE cur; END;</pre>

Umwandeln von Oracle in Amazon RDS für Oracle

Bei der Migration von Oracle-Schema und -Code in Amazon RDS für Oracle sind einige Punkte zu beachten:

- AWS SCT kann Verzeichnisobjekte zum Objektbaum hinzufügen. Verzeichnisobjekte sind logische Strukturen, die jeweils ein physisches Verzeichnis im Dateisystem des Servers darstellen. Sie können Verzeichnisobjekte mit Paketen, wie z. B. DBMS_LOB, UTL_FILE, DBMS_FILE_TRANSFER, dem DATAPUMP Dienstprogramm usw. verwenden.
- AWS SCT unterstützt die Konvertierung von Oracle-Tablespaces in eine Amazon RDS for Oracle DB-Instance. Oracle speichert Daten logisch in Tabellenräumen und physisch in Datendateien, die dem entsprechenden Tabellenraum zugeordnet sind. In Oracle können Sie einen Tabellenraum mit Datendateinamen erstellen. Amazon RDS unterstützt Oracle Managed Files (OMF) nur für Datendateien, Protokolldateien und Kontrolldateien. AWS SCT erstellt die benötigten Datendateien während der Konvertierung.
- AWS SCT kann Rollen und Rechte auf Serverebene konvertieren. Die Oracle-Datenbank-Engine verwendet rollenbasierte Sicherheit. Eine Rolle ist eine Sammlung von Sonderrechten, die Sie einem Benutzer gewähren oder entziehen können. Eine vordefinierte Rolle in Amazon RDS mit dem Namen DBA lässt normalerweise alle administrativen Sonderrechte in einer Oracle-Datenbank-Engine zu. Die folgenden Sonderrechte sind nicht für die Feedback-Schleifen-Rolle in einer Amazon RDS-DB-Instance verfügbar, welche die Oracle-Engine verwendet:
 - Ändern der Datenbank

- Ändern des Systems
- Erstellen eines Verzeichnisses
- Gewähren von Sonderrechten
- Gewähren einer Rolle
- Erstellen eines externen Auftrags

Sie können alle anderen Berechtigungen für eine Amazon RDS for Oracle-Benutzerrolle, u. a. erweiterte Filterungs- und Spaltenberechtigungen, erteilen.

- AWS SCT unterstützt die Konvertierung von Oracle-Jobs in Jobs, die auf Amazon RDS for Oracle ausgeführt werden können. Es gibt einige Einschränkungen für die Konvertierung, unter anderem:
 - Ausführbare Aufträge werden nicht unterstützt.
 - Die Einplanung von Aufträgen, die den ANYDATA-Datentyp als Argument verwenden, werden nicht unterstützt.
- Oracle Real Application Clusters (RAC) One Node ist eine Option der Oracle Database Enterprise Edition, die mit Oracle Database 11g Release 2 eingeführt wurde. Amazon RDS for Oracle unterstützt die RAC-Funktion nicht. Verwenden Sie Amazon RDS Multi-AZ für Hochverfügbarkeit.

Bei einer Multi-AZ-Bereitstellung sorgt Amazon RDS für eine automatische Bereitstellung und Verwaltung eines synchronen Standby-Replikats in einer anderen Availability Zone. Die primäre DB-Instance wird über die Availability Zone synchron auf eine Standby-Replikation repliziert. Diese Funktionalität sorgt für Datenredundanz, vermeidet das Einfrieren von E/A-Vorgängen und minimiert Latenzspitzen während der Durchführung von Systemsicherungen.

- Oracle Spatial stellt ein SQL-Schema und Funktionen bereit, die Speicherung, Abruf, Update und Abfrage von Sammlungen großer Datensätze in einer Oracle-Datenbank ermöglichen. Oracle Locator stellt Funktionalitäten zur Verfügung, die üblicherweise für die Unterstützung von Anwendungen auf Internet- und Wireless-Servicebasis sowie von GIS-Lösungen auf Partnerbasis erforderlich sind. Oracle Locator ist ein begrenzter Teilbereich von Oracle Spatial.

Um Oracle Spatial- und Oracle Locator-Funktionen verwenden zu können, fügen Sie die Option SPATIAL oder die Option LOCATOR (schließen sich gegenseitig aus) zur Optionsgruppe Ihrer DB-Instance hinzu.

Es gibt einige Voraussetzungen für die Verwendung von Oracle Spatial und Oracle Locator auf einer Amazon RDS for Oracle-DB-Instance:

- Die Instanz sollte Oracle Enterprise Edition Version 12.1.0.2.v6 oder höher oder 11.2.0.4.v10 oder höher verwenden.

- Die Instance sollte sich innerhalb einer Virtual Private Cloud (VPC) befinden.
- Die Instance sollte die Version der DB-Instance-Klasse sein, die die Oracle-Funktion unterstützen kann. Beispiel: Oracle Spatial wird für die DB-Instance-Klassen db.m1.small, db.t1.micro, db.t2.micro und db.t2.small nicht unterstützt. Weitere Informationen finden Sie unter [DB-Instance-Klassenunterstützung für Oracle](#).
- Die Instance muss die Option „Auto Minor Version Upgrade“ (Upgrade einer Unterversion automatisch durchführen) aktiviert haben. Amazon RDS aktualisiert Ihre DB-Instance auf die neueste Oracle-PSU, falls Schwachstellen mit einem CVSS-Schweregrad von 9+ oder andere gemeldete Schwachstellen vorliegen. Weitere Informationen finden Sie unter [Einstellungen für Oracle-DB-Instances](#).
- Wenn Ihre DB-Instance Version 11.2.0.4.v10 oder höher ist, müssen Sie die XMLDB-Option installieren. Weitere Informationen finden Sie unter [Oracle-XML-Datenbank](#).
- Sie sollten über eine Oracle Spatial-Lizenz von Oracle verfügen. Weitere Informationen finden Sie unter [Oracle Spatial and Graph](#) in der Oracle-Dokumentation.
- Data Guard ist in der Oracle Database Enterprise Edition enthalten. Verwenden Sie Amazon RDS Multi-AZ für Hochverfügbarkeit.

Bei einer Multi-AZ-Bereitstellung sorgt Amazon RDS für eine automatische Bereitstellung und Verwaltung eines synchronen Standby-Replikats in einer anderen Availability Zone. Die primäre DB-Instance wird über die Availability Zone synchron auf eine Standby-Replikation repliziert. Diese Funktionalität sorgt für Datenredundanz, vermeidet das Einfrieren von E/A-Vorgängen und minimiert Latenzspitzen während der Durchführung von Systemsicherungen.

- AWS SCT unterstützt die Konvertierung von Oracle DBMS_SCHEDULER-Objekten bei der Migration zu Amazon RDS for Oracle. Der AWS SCT Bewertungsbericht gibt an, ob ein Zeitplanobjekt konvertiert werden kann. Weitere Informationen zur Verwendung von Zeitplanobjekten mit Amazon RDS finden Sie in der [Amazon RDS-Dokumentation](#).
- Für Umwandlungen von Oracle zu Amazon RDS wird DB Links unterstützt. Ein Datenbanklink ist ein Schemaobjekt in einer Datenbank, mit dessen Hilfe Sie auf Objekte in einer anderen Datenbank zugreifen können. Die andere Datenbank muss keine Oracle-Datenbank sein. Für den Zugriff auf andere als Oracle-Datenbanken müssen Sie jedoch Oracle Heterogeneous Services verwenden.

Nachdem Sie einen Datenbanklink erstellt haben, können Sie den Link in SQL-Anweisungen verwenden, um auf Tabellen, Ansichten und PL/SQL-Objekte in der anderen Datenbank zu

verweisen. Um einen Datenbanklink zu verwenden, fügen Sie dem Tabellen-, Ansicht- oder PL/SQL-Objektnamen `@dblink` hinzu. Sie können eine Tabelle oder Ansicht in der anderen Datenbank mit der SELECT-Anweisung abfragen. Weitere Informationen zur Verwendung von Oracle-Datenbanklinks finden Sie in der [Oracle-Dokumentation](#).

Weitere Informationen zur Verwendung von Datenbanklinks mit Amazon RDS finden Sie in der [Amazon RDS-Dokumentation](#).

- Der AWS SCT Bewertungsbericht enthält Servermetriken für die Konvertierung. Zu diesen Metriken über Ihre Oracle-Instance gehören:
 - Datenverarbeitungs- und Speicherkapazität der Ziel-DB-Instance.
 - Nicht unterstützte Oracle-Funktionen wie Real Application Clusters, die Amazon RDS nicht unterstützt.
 - Datenträger-Lese-/Schreib-Last
 - Durchschnittlicher Datenträger-Gesamtdurchsatz
 - Serverinformationen wie Servername, Betriebssystem, Hostname und Zeichensatz.

Rechte für RDS for Oracle als Ziel

Um zu Amazon RDS for Oracle zu migrieren, erstellen Sie einen privilegierten Datenbankbenutzer. Sie können das folgende Codebeispiel verwenden.

```
CREATE USER user_name IDENTIFIED BY your_password;  
  
-- System privileges  
GRANT DROP ANY CUBE BUILD PROCESS TO user_name;  
GRANT ALTER ANY CUBE TO user_name;  
GRANT CREATE ANY CUBE DIMENSION TO user_name;  
GRANT CREATE ANY ASSEMBLY TO user_name;  
GRANT ALTER ANY RULE TO user_name;  
GRANT SELECT ANY DICTIONARY TO user_name;  
GRANT ALTER ANY DIMENSION TO user_name;  
GRANT CREATE ANY DIMENSION TO user_name;  
GRANT ALTER ANY TYPE TO user_name;  
GRANT DROP ANY TRIGGER TO user_name;  
GRANT CREATE ANY VIEW TO user_name;  
GRANT ALTER ANY CUBE BUILD PROCESS TO user_name;  
GRANT CREATE ANY CREDENTIAL TO user_name;  
GRANT DROP ANY CUBE DIMENSION TO user_name;  
GRANT DROP ANY ASSEMBLY TO user_name;
```

```
GRANT DROP ANY PROCEDURE TO user_name;  
GRANT ALTER ANY PROCEDURE TO user_name;  
GRANT ALTER ANY SQL TRANSLATION PROFILE TO user_name;  
GRANT DROP ANY MEASURE FOLDER TO user_name;  
GRANT CREATE ANY MEASURE FOLDER TO user_name;  
GRANT DROP ANY CUBE TO user_name;  
GRANT DROP ANY MINING MODEL TO user_name;  
GRANT CREATE ANY MINING MODEL TO user_name;  
GRANT DROP ANY EDITION TO user_name;  
GRANT CREATE ANY EVALUATION CONTEXT TO user_name;  
GRANT DROP ANY DIMENSION TO user_name;  
GRANT ALTER ANY INDEXTYPE TO user_name;  
GRANT DROP ANY TYPE TO user_name;  
GRANT CREATE ANY PROCEDURE TO user_name;  
GRANT CREATE ANY SQL TRANSLATION PROFILE TO user_name;  
GRANT CREATE ANY CUBE TO user_name;  
GRANT COMMENT ANY MINING MODEL TO user_name;  
GRANT ALTER ANY MINING MODEL TO user_name;  
GRANT DROP ANY SQL PROFILE TO user_name;  
GRANT CREATE ANY JOB TO user_name;  
GRANT DROP ANY EVALUATION CONTEXT TO user_name;  
GRANT ALTER ANY EVALUATION CONTEXT TO user_name;  
GRANT CREATE ANY INDEXTYPE TO user_name;  
GRANT CREATE ANY OPERATOR TO user_name;  
GRANT CREATE ANY TRIGGER TO user_name;  
GRANT DROP ANY ROLE TO user_name;  
GRANT DROP ANY SEQUENCE TO user_name;  
GRANT DROP ANY CLUSTER TO user_name;  
GRANT DROP ANY SQL TRANSLATION PROFILE TO user_name;  
GRANT ALTER ANY ASSEMBLY TO user_name;  
GRANT CREATE ANY RULE SET TO user_name;  
GRANT ALTER ANY OUTLINE TO user_name;  
GRANT UNDER ANY TYPE TO user_name;  
GRANT CREATE ANY TYPE TO user_name;  
GRANT DROP ANY MATERIALIZED VIEW TO user_name;  
GRANT ALTER ANY ROLE TO user_name;  
GRANT DROP ANY VIEW TO user_name;  
GRANT ALTER ANY INDEX TO user_name;  
GRANT COMMENT ANY TABLE TO user_name;  
GRANT CREATE ANY TABLE TO user_name;  
GRANT CREATE USER TO user_name;  
GRANT DROP ANY RULE SET TO user_name;  
GRANT CREATE ANY CONTEXT TO user_name;  
GRANT DROP ANY INDEXTYPE TO user_name;
```

```
GRANT ALTER ANY OPERATOR TO user_name;  
GRANT CREATE ANY MATERIALIZED VIEW TO user_name;  
GRANT ALTER ANY SEQUENCE TO user_name;  
GRANT DROP ANY SYNONYM TO user_name;  
GRANT CREATE ANY SYNONYM TO user_name;  
GRANT DROP USER TO user_name;  
GRANT ALTER ANY MEASURE FOLDER TO user_name;  
GRANT ALTER ANY EDITION TO user_name;  
GRANT DROP ANY RULE TO user_name;  
GRANT CREATE ANY RULE TO user_name;  
GRANT ALTER ANY RULE SET TO user_name;  
GRANT CREATE ANY OUTLINE TO user_name;  
GRANT UNDER ANY TABLE TO user_name;  
GRANT UNDER ANY VIEW TO user_name;  
GRANT DROP ANY DIRECTORY TO user_name;  
GRANT ALTER ANY CLUSTER TO user_name;  
GRANT CREATE ANY CLUSTER TO user_name;  
GRANT ALTER ANY TABLE TO user_name;  
GRANT CREATE ANY CUBE BUILD PROCESS TO user_name;  
GRANT ALTER ANY CUBE DIMENSION TO user_name;  
GRANT CREATE ANY EDITION TO user_name;  
GRANT CREATE ANY SQL PROFILE TO user_name;  
GRANT ALTER ANY SQL PROFILE TO user_name;  
GRANT DROP ANY OUTLINE TO user_name;  
GRANT DROP ANY CONTEXT TO user_name;  
GRANT DROP ANY OPERATOR TO user_name;  
GRANT DROP ANY LIBRARY TO user_name;  
GRANT ALTER ANY LIBRARY TO user_name;  
GRANT CREATE ANY LIBRARY TO user_name;  
GRANT ALTER ANY MATERIALIZED VIEW TO user_name;  
GRANT ALTER ANY TRIGGER TO user_name;  
GRANT CREATE ANY SEQUENCE TO user_name;  
GRANT DROP ANY INDEX TO user_name;  
GRANT CREATE ANY INDEX TO user_name;  
GRANT DROP ANY TABLE TO user_name;  
GRANT SELECT_CATALOG_ROLE TO user_name;  
GRANT SELECT ANY SEQUENCE TO user_name;  
  
-- Database Links  
GRANT CREATE DATABASE LINK TO user_name;  
GRANT CREATE PUBLIC DATABASE LINK TO user_name;  
GRANT DROP PUBLIC DATABASE LINK TO user_name;
```

```
-- Server Level Objects (directory)
GRANT CREATE ANY DIRECTORY TO user_name;
GRANT DROP ANY DIRECTORY TO user_name;
-- (for RDS only)
GRANT EXECUTE ON RDSADMIN.RDSADMIN_UTIL TO user_name;

-- Server Level Objects (tablespace)
GRANT CREATE TABLESPACE TO user_name;
GRANT DROP TABLESPACE TO user_name;

-- Server Level Objects (user roles)
/* (grant source privileges with admin option or convert roles/privs as DBA) */

-- Queues
grant execute on DBMS_AQADM to user_name;
grant aq_administrator_role to user_name;

-- for Materialized View Logs creation
GRANT SELECT ANY TABLE TO user_name;

-- Roles
GRANT RESOURCE TO user_name;
GRANT CONNECT TO user_name;
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen *Sie dann your_password* durch ein sicheres Passwort.

Einschränkungen bei der Konvertierung von Oracle zu Amazon RDS for Oracle

Bei der Migration von Oracle-Schema und -Code in Amazon RDS for Oracle sind einige Einschränkungen zu beachten:

- Eine vordefinierte Rolle in Amazon RDS mit den Namen DBA lässt normalerweise alle administrativen Sonderrechte in einer Oracle-Datenbank-Engine zu. Die folgenden Sonderrechte sind nicht für die Feedback-Schleifen-Rolle in einer Amazon RDS-DB-Instance verfügbar, welche die Oracle-Engine verwendet:
 - Ändern der Datenbank
 - Ändern des Systems
 - Erstellen eines Verzeichnisses
 - Gewähren von Sonderrechten

- Gewähren einer Rolle
- Erstellen eines externen Auftrags

Sie können einer Oracle RDS-Benutzerrolle alle anderen Berechtigungen erteilen.

- Amazon RDS for Oracle unterstützt herkömmliches Auditing, feinkörniges Auditing mithilfe des DBMS_FGA-Pakets und Oracle Unified Auditing.
- Amazon RDS for Oracle unterstützt Change Data Capture (CDC) nicht. Um CDC während und nach einer Datenbankmigration durchzuführen, verwenden Sie AWS Database Migration Service.

Verwendung von PostgreSQL als Quelle für AWS SCT

Sie können AWS SCT verwenden, um Schemas, Datenbankcodeobjekte und Anwendungscode von PostgreSQL in die folgenden Ziele zu konvertieren:

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition

Weitere Informationen finden Sie in den folgenden Abschnitten:

Themen

- [Rechte für PostgreSQL als Quelldatenbank](#)
- [Herstellen einer Verbindung zu PostgreSQL als Quelle](#)
- [Rechte für MySQL als Zieldatenbank](#)

Rechte für PostgreSQL als Quelldatenbank

Die für PostgreSQL als Quelle erforderlichen Rechte lauten wie folgt:

- CONNECT ON DATABASE *<database_name>*
- USAGE ON SCHEMA *<database_name>*
- SELECT ON ALL TABLES IN SCHEMA *<database_name>*
- SELECT ON ALL SEQUENCES IN SCHEMA *<database_name>*

Herstellen einer Verbindung zu PostgreSQL als Quelle

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer PostgreSQL-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer PostgreSQL-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen.
2. Wählen Sie PostgreSQL und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für die Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWS Secret den Namen des Geheimnisses aus.
 2. Wählen Sie Füllen, um automatisch alle Werte im Dialogfeld für die Datenbankverbindung aus Secrets Manager einzugeben.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen der PostgreSQL-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	<p>Geben Sie den Domain Name System (DNS) Namen oder die IP-Adresse Ihres Quell-Datenbankservers ein.</p> <p>Sie können eine Verbindung zu Ihrer PostgreSQL-Datenbank mit einem IPv6-Adressprotokoll bereitstellen. Stellen Sie dafür sicher, dass Sie für die Eingabe der IP-Adresse eckige Klammern verwenden, wie im folgenden Beispiel gezeigt.</p>

Parameter	Action
	<div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; display: inline-block;">[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</div>
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der PostgreSQL-Datenbank ein.
User name und Password	<p>Geben Sie die Datenbankanmeldeinformationen ein, um eine Verbindung mit Ihrem Quell-Datenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie in einem Projekt eine Verbindung zu Ihrer Datenbank herstellen möchten. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert es das Passwort standardmäßig nicht. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>
Use SSL	<p>Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden. Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL ein:</p> <ul style="list-style-type: none">• Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Truststores zu überprüfen.• Trust Store: Der Standort eines Truststores, der Zertifikate enthält. Damit dieser Standort im Bereich Allgemeine Einstellungen angezeigt wird, müssen Sie ihn unbedingt hinzufügen.

Parameter	Action
Passwort speichern	AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankpasswort speichern und schnell auf die Datenbank zugreifen, ohne dass Sie das Passwort eingeben müssen.
PostgreSQL-Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projektinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie **Verbindung testen**, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden kann.
6. Wählen Sie **Connect**, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Rechte für MySQL als Zieldatenbank

Für MySQL als Ziel sind bei der Migration von PostgreSQL die folgenden Rechte erforderlich:

- ERSTELLEN AM * . *
- ÄNDERN AM * . *
- KOMM VORBEI * . *
- INDEX AUF* . *
- REFERENZEN AUF* . *
- SELECT ON *.*
- ANSICHT ERSTELLEN AUF* . *
- SHOW VIEW ON *.*
- AUSLÖSER AM * . *
- ROUTINE ERSTELLEN AM * . *

- ROUTINE ÄNDERN AM *.*
- AUSFÜHREN AM *.*
- EINFÜGEN, AUF AWS_POSTGRESQL_EXT AKTUALISIEREN.*
- EINFÜGEN, AKTUALISIEREN, LÖSCHEN AUF AWS_POSTGRESQL_EXT_DATA.*
- ERSTELLE TEMPORÄRE TABELLEN AUF AWS_POSTGRESQL_EXT_DATA.*

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT INSERT, UPDATE ON AWS_POSTGRESQL_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_POSTGRESQL_EXT_DATA.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_POSTGRESQL_EXT_DATA.* TO 'user_name';
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen *Sie dann your_password* durch ein sicheres Passwort.

Um Amazon RDS for MySQL oder Aurora MySQL als Ziel zu verwenden, legen Sie den `lower_case_table_names` Parameter auf `1`. Dieser Wert bedeutet, dass der MySQL-Server Bezeichner solcher Objektnamen wie Tabellen, Indizes, Trigger und Datenbanken ohne Berücksichtigung von Groß- und Kleinschreibung behandelt. Wenn Sie die binäre Protokollierung in Ihrer Zielinstantz aktiviert haben, setzen Sie den `log_bin_trust_function_creators` Parameter auf `1`. In diesem Fall müssen Sie die `NO SQL Merkmale READS SQL DATA` oder nicht verwenden `DETERMINISTIC`, um gespeicherte Funktionen zu erstellen. Um diese Parameter zu konfigurieren, erstellen Sie eine neue DB-Parametergruppe oder ändern Sie eine vorhandene DB-Parametergruppe.

Verwendung von SAP ASE (Sybase ASE) als Quelle für AWS SCT

Sie können AWS SCT verwenden, um Schemas, Datenbankcode-Objekte und Anwendungscode von SAP (Sybase) Adaptive Server Enterprise (ASE) in die folgenden Ziele zu konvertieren:

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for MariaDB
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition

Weitere Informationen finden Sie in den folgenden Abschnitten:

Themen

- [Privilegien für SAP ASE als Quelldatenbank](#)
- [Verbindung zu SAP ASE \(Sybase\) als Quelle herstellen](#)
- [Rechte für MySQL als Zieldatenbank](#)
- [Einstellungen für die Konvertierung von SAP ASE zu MySQL](#)
- [Privilegien für PostgreSQL als Zieldatenbank](#)
- [Einstellungen für die Konvertierung von SAP ASE zu PostgreSQL](#)

Privilegien für SAP ASE als Quelldatenbank

Um eine SAP ASE-Datenbank als Quelle zu verwenden, erstellen Sie einen Datenbankbenutzer und gewähren Berechtigungen. Führen Sie dazu die folgenden Schritte aus.

Erstellen und Konfigurieren eines Datenbankbenutzers

1. Stellt eine Verbindung zur Quelldatenbank her.
2. Erstellen Sie einen Datenbankbenutzer mit den folgenden Befehlen. Geben Sie ein Passwort für den neuen Benutzer ein.

```
USE master
CREATE LOGIN min_privs WITH PASSWORD <password>
sp_adduser min_privs
grant select on dbo.spt_values to min_privs
```

```
grant select on asehostname to min_privs
```

3. Gewähren Sie für jede Datenbank, die Sie migrieren möchten, die folgenden Rechte.

```
USE <database_name>
sp_adduser min_privs
grant select on dbo.sysusers to min_privs
grant select on dbo.sysobjects to min_privs
grant select on dbo.sysindexes to min_privs
grant select on dbo.syscolumns to min_privs
grant select on dbo.sysreferences to min_privs
grant select on dbo.syscomments to min_privs
grant select on dbo.syspartitions to min_privs
grant select on dbo.syspartitionkeys to min_privs
grant select on dbo.sysconstraints to min_privs
grant select on dbo.systypes to min_privs
grant select on dbo.sysqueryplans to min_privs
```

Verbindung zu SAP ASE (Sybase) als Quelle herstellen

Verwenden Sie die folgende Vorgehensweise, um per eine Verbindung mit Ihrer SAP ASE-Quelldatenbank herzustellenAWS Schema Conversion Tool.

So stellen Sie eine Verbindung zu einer SAP ASE-Quelldatenbank her


1. Wählen Sie in derAWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie SAP ASE und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für die Datenbank ein. AWS SCTzeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Datenbankmeldeinformationen vonAWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWS Secret den Namen des Geheims aus.
 2. Wählen Sie Füllen, um automatisch alle Werte im Dialogfeld für die Datenbankverbindung aus Secrets Manager einzugeben.

Hinweise zur Verwendung von Datenbankanmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen der SAP ASE-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	Geben Sie den Domain Name System (DNS) -Namen oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der SAP ASE-Datenbank ein.
User name und Password	<p>Geben Sie die Datenbankanmeldeinformationen ein, um eine Verbindung mit Ihrem Quelldatenbankserver herzustellen.</p> <div data-bbox="656 999 1510 1602" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> Note</p><p>AWS SCT verwendet das Passwort nur, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie in einem Projekt eine Verbindung zu Ihrer Datenbank herstellen möchten. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, speichert es das Passwort standardmäßig nicht. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p></div>

Parameter	Action
Use SSL	<p>Wählen Sie diese Option aus, um per Secure Sockets Layer (SSL) eine Verbindung mit Ihrer Datenbank herzustellen. Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL ein:</p> <ul style="list-style-type: none"> • Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Truststores zu überprüfen. • Trust Store: Der Standort eines Truststores, der Zertifikate enthält.
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankpasswort speichern und schnell auf die Datenbank zugreifen, ohne dass Sie das Passwort eingeben müssen.</p>
SAP ASE-Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projektinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden kann.
6. Wählen Sie Connect, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Rechte für MySQL als Zieldatenbank

Die für MySQL als Ziel erforderlichen Rechte lauten wie folgt:

- ERSTELLEN AM * . *

- ÄNDERN SIE AM . *
- KOMM VORBEI AM * . *
- INDEX AUF . *
- REFERENZEN AUF . *
- SELECT ON *.*
- ANSICHT ERSTELLEN AUF* . *
- SHOW VIEW ON *.*
- AUSLÖSER AM . *
- ROUTINE ERSTELLEN AM * . *
- ROUTINE ÄNDERN AM * . *
- AUSFÜHREN AM * . *
- EINFÜGEN, AUF AWS_SAPASE_EXT AKTUALISIEREN. *
- ERSTELLE TEMPORÄRE TABELLEN AUF AWS_SAPASE_EXT. *

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT INSERT, UPDATE ON AWS_SAPASE_EXT.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_SAPASE_EXT.* TO 'user_name';
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen *Sie dann your_password* durch ein sicheres Passwort.

Um Amazon RDS for MySQL oder Aurora MySQL als Ziel zu verwenden, legen Sie den `lower_case_table_names` Parameter auf `1`. Dieser Wert bedeutet, dass der MySQL-Server Bezeichner solcher Objektnamen wie Tabellen, Indizes, Trigger und Datenbanken ohne Berücksichtigung von Groß- und Kleinschreibung behandelt. Wenn Sie die binäre Protokollierung in Ihrer Zielinstanz aktiviert haben, setzen Sie den `log_bin_trust_function_creators` Parameter auf `1`. In diesem Fall müssen Sie die `NO SQL` Merkmale `READS SQL DATA` oder nicht verwenden `DETERMINISTIC`, um gespeicherte Funktionen zu erstellen. Um diese Parameter zu stellen, legen Sie eine neue DB-Parametergruppe an oder ändern Sie eine vorhandene DB-Parametergruppe.

Einstellungen für die Konvertierung von SAP ASE zu MySQL

Um die Konvertierungseinstellungen von SAP ASE zu MySQL zu bearbeiten, wählen Sie Einstellungen und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste SAP ASE und dann SAP ASE — MySQL oder SAP ASE — Amazon Aurora (MySQL-kompatibel). AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von SAP ASE zu PostgreSQL an.

Die Konvertierungseinstellungen von SAP ASE zu MySQL in AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionspunkten im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare zum konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und einem höheren Schweregrad hinzufügen den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte des ausgewählten Schweregrads und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler. Um Kommentare für alle Aktionselemente in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die genauen Namen der Quelldatenbankobjekte im konvertierten Code zu verwenden.

AWS SCT konvertiert standardmäßig die Namen von Datenbankobjekten, Variablen und Parametern in Kleinbuchstaben. Um die ursprüngliche Groß- und Kleinschreibung für diese Namen beizubehalten, wählen Sie Quelldatenbankobjektnamen als Groß- und Kleinschreibung behandeln aus. Wählen Sie diese Option, wenn Sie in Ihrem SAP ASE-Quelldatenbankserver Objektnamen verwenden, bei denen zwischen Groß- und Kleinschreibung unterschieden wird.

Privilegien für PostgreSQL als Zieldatenbank

Um PostgreSQL als Ziel zu verwenden, AWS SCT ist das `CREATE ON DATABASE` Privileg erforderlich. Stellen Sie sicher, dass Sie dieses Privileg für jede PostgreSQL-Zieldatenbank gewähren.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in `"$user", public_synonyms, public`.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';  
GRANT CREATE ON DATABASE db_name TO user_name;  
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Ersetzen Sie im vorherigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie dann *db_name* durch den Namen Ihrer Zieldatenbank. Ersetzen *Sie abschließend your_password* durch ein sicheres Passwort.

In PostgreSQL `superuser` kann nur der Schemabesitzer oder ein Schemabesitzer ein Schema löschen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Besitzer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, können Sie eine Fehlermeldung erhalten, wenn ein Schema nicht gelöscht werden kann. Verwenden Sie die `superuser` Rolle, um diese Fehlermeldung zu vermeiden.

Einstellungen für die Konvertierung von SAP ASE zu PostgreSQL

Um die Konvertierungseinstellungen von SAP ASE zu PostgreSQL zu bearbeiten, wählen Sie Einstellungen und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste SAP ASE und dann SAP ASE — PostgreSQL oder SAP ASE — Amazon Aurora (PostgreSQL-kompatibel). AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von SAP ASE zu PostgreSQL an.

Die Konvertierungseinstellungen von SAP ASE zu PostgreSQL in AWS SCT enthalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionspunkten im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare zum konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und einem höheren Schweregrad hinzufügen den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte des ausgewählten Schweregrads und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler. Um Kommentare für alle Aktionselemente in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die Vorlage zu definieren, die für die Schemanamen im konvertierten Code verwendet werden soll. Wählen Sie unter Vorlage zur Generierung von Schemanamen eine der folgenden Optionen aus:
 - `<source_db>`— Verwendet den SAP ASE-Datenbanknamen als Schemanamen in PostgreSQL.
 - `<source_schema>`— Verwendet den SAP ASE-Schemanamen als Schemanamen in PostgreSQL.
 - `_ <source_db><schema>`— Verwendet eine Kombination aus SAP ASE-Datenbank und Schemanamen als Schemanamen in PostgreSQL.
- Um die genauen Namen der Quelldatenbankobjekte im konvertierten Code zu verwenden.

AWS SCT konvertiert standardmäßig die Namen von Datenbankobjekten, Variablen und Parametern in Kleinbuchstaben. Um die ursprüngliche Groß- und Kleinschreibung für diese Namen beizubehalten, wählen Sie Quelldatenbankobjektnamen als Groß- und Kleinschreibung behandeln aus. Wählen Sie diese Option, wenn Sie in Ihrem SAP ASE-Quelldatenbankserver Objektnamen verwenden, bei denen zwischen Groß- und Kleinschreibung unterschieden wird.

Bei Operationen, bei denen die Groß- und Kleinschreibung beachtet wird, AWS SCT kann die Konvertierung von Datenbankobjektnamen in Kleinbuchstaben vermieden werden. Wählen Sie dazu für Operationen, bei denen zwischen Groß- und Kleinschreibung unterschieden wird, die Option Umwandlung in Kleinbuchstaben vermeiden.

- Um die Verwendung von Indizes mit demselben Namen in verschiedenen Tabellen in SAP ASE zu ermöglichen.

In PostgreSQL müssen alle Indexnamen, die Sie im Schema verwenden, eindeutig sein. Um sicherzustellen, dass eindeutige Namen für all Ihre Indizes AWS SCT generiert werden, wählen Sie Eindeutige Namen für Indizes generieren aus.

Verwenden von Microsoft SQL Server als Quelle für AWS SCT

Sie können AWS SCT um Schemas, Datenbankcodeobjekte und Anwendungscode aus SQL Server in die folgenden Ziele zu konvertieren:

- Amazon RDS for MySQL
- Amazon Aurora MySQL-Compatible Edition
- Amazon RDS for PostgreSQL
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon RDS for SQL Server
- Amazon RDS for MariaDB

Note

AWS SCT unterstützt nicht die Verwendung von Amazon RDS für SQL Server als Quelle.

Sie können AWS SCT um einen Bewertungsbericht für die Migration von Schemas, Datenbankcodeobjekten und Anwendungscode von SQL Server zu Babelfish for Aurora PostgreSQL zu erstellen, wie im Folgenden beschrieben.

Themen

- [Rechte für die Microsoft SQL -Prozedur als -Quelle.](#)
- [Verwenden der Windows-Authentifizierung bei Verwendung von Microsoft SQL Server als Quelle](#)
- [Verbindung mit SQL Server als -Quelle herstellen.](#)
- [Umwandeln von SQL Server in MySQL](#)
- [Umwandeln von SQL Server in PostgreSQL](#)
- [Umwandeln von SQL Server in Amazon RDS für SQL Server](#)

Rechte für die Microsoft SQL -Prozedur als -Quelle.

Die für Microsoft SQL Server als Quelle erforderlichen Rechte lauten wie folgt:

- VIEW DEFINITION
- VIEW DATABASE STATE

Das `VIEW DEFINITION` Mit dieser Berechtigung können Benutzer mit öffentlichem Zugriff Objektdefinitionen einsehen. AWS SCT verwendet die `VIEW DATABASE STATE` das Recht, die Funktionen der SQL Server Enterprise Edition zu überprüfen.

Gewähren Sie die erforderlichen Berechtigungen für alle Datenbanken, deren Schemata Sie konvertieren.

Erteilen Sie zusätzlich die folgenden Rechte für `master` Datenbank:

- `VIEW SERVER STATE`
- `EINE BELIEBIGE DEFINITION ANSEHEN`

AWS SCT verwendet die `VIEW SERVER STATE` das Recht, Servereinstellungen und -konfigurationen zu sammeln. Stellen Sie sicher, dass Sie die `VIEW ANY DEFINITION` das Recht, Endpunkte einzusehen.

Um Informationen zu Microsoft Analysis Services zu lesen, führen Sie den folgenden Befehl auf der `master` Datenbank.

```
EXEC master..sp_addsrvrolemember @loginame = N'<user_name>', @rolename = N'sysadmin'
```

Im vorherigen Beispiel ersetzen Sie die `-Prozedur.<user_name>` Platzhalter mit dem Namen des Benutzers, dem Sie zuvor die Berechtigungen erteilt haben.

Um Informationen über den SQL Server-Agent zu lesen, fügen Sie Ihren Benutzer zur `SQLAgentUser` Rolle. Führen Sie die `-Prozedur` aus `msdb` Datenbank.

```
EXEC sp_addrolemember <SQLAgentRole>, <user_name>;
```

Im vorherigen Beispiel ersetzen Sie die `-Prozedur.<SQLAgentRole>` Platzhalter mit dem Namen der SQL Server-Agent-Rolle. Ersetzen Sie dann den `<user_name>` Platzhalter mit dem Namen des Benutzers, dem Sie zuvor die Rechte erteilt haben. Weitere Informationen finden Sie unter [Hinzufügen eines Benutzers zum SQLAgentUser Rolle](#) in der Amazon RDS-Benutzerhandbuch.

Um den Versand von Protokollen zu erkennen, gewähren Sie `SELECT on dbo.log_shipping_primary_databases` Privileg auf `msdb` Datenbank.

Um den Benachrichtigungsansatz der DDL-Replikation zu verwenden, gewähren Sie `RECEIVE ON <schema_name>.<queue_name>` Rechte für Ihre Quelldatenbanken. Ersetzen Sie in diesem

Beispiel die `<schema_name>` Platzhalter mit der -Prozedur Ihrer -Prozedur. Ersetzen Sie dann den `<queue_name>` Platzhalter mit dem Namen einer Queue-Tabelle.

Verwenden der Windows-Authentifizierung bei Verwendung von Microsoft SQL Server als Quelle

Wenn Ihre Anwendung auf einem Windows-basierten Intranet ausgeführt wird, können Sie möglicherweise die Windows-Authentifizierung für den Zugriff auf die Datenbank verwenden. Die Windows-Authentifizierung verwendet die aktuelle Windows-Identität auf der Betriebssystem-Thread für den Zugriff auf die SQL Server-Datenbank. Sie können dann die Windows-Identität auf eine SQL Server-Datenbank und Berechtigungen abbilden. Zum Herstellen einer Verbindung mit SQL Server unter Verwendung der Windows-Authentifizierung müssen Sie die Windows-Identität angeben, die Ihre Anwendung verwendet. Außerdem müssen Sie der Windows-Identität Zugriff auf die SQL Server-Datenbank erteilen.

SQL Server verfügt über zwei Zugriffsmodi: Windows-Authentifizierungs-Modus und gemischter Modus. Der Windows-Authentifizierungs-Modus aktiviert die Windows-Authentifizierung und deaktiviert die SQL Server-Authentifizierung. Der gemischte Modus aktiviert die Windows-Authentifizierung und die SQL Server-Authentifizierung. Die Windows-Authentifizierung ist immer verfügbar und kann nicht deaktiviert werden. Weitere Informationen über die Windows-Authentifizierung finden Sie in der Microsoft Windows-Dokumentation.

Nachfolgend finden Sie ein Beispiel für das Erstellen eines Benutzers in TEST_DB.

```
USE [TEST_DB]
CREATE USER [TestUser] FOR LOGIN [TestDomain\TestUser]
GRANT VIEW DEFINITION TO [TestUser]
GRANT VIEW DATABASE STATE TO [TestUser]
```

Die Windows-Authentifizierung über die JDBC-Verbindung verwenden.

Der JDBC-Treiber unterstützt keine Windows-Authentifizierung, wenn der Treiber für Nicht-Windows-Betriebssysteme verwendet wird. Anmeldeinformationen für die Windows-Authentifizierung, wie z. B. Benutzername und Kennwort, werden nicht automatisch angegeben, wenn von einem anderen Betriebssystem als Windows aus eine Verbindung zu SQL Server hergestellt wird. In solchen Fällen müssen die Anwendungen stattdessen die SQL Server-Authentifizierung verwenden.

In der JDBC-Verbindungszeichenfolge, muss der Parameter `integratedSecurity` angegeben werden, um eine Verbindung unter Verwendung der Windows-Authentifizierung herzustellen. Der

JDBC-Treiber unterstützt die integrierte Windows-Authentifizierung auf Windows-Betriebssysteme über den `integratedSecurity`-Verbindungszeichenfolgenparameter.

Verwendung der integrierten Authentifizierung

1. Installieren Sie den JDBC-Treiber.
2. Kopieren Sie die `sqljdbc_auth.dll`-Datei in ein Verzeichnis im Windows-Systempfad auf dem Computer, auf dem der JDBC-Treiber installiert ist.

Das `sqljdbc_auth.dll` Die Dateien sind am folgenden Speicherort installiert:

```
<Installationsverzeichnis>\sqljdbc_<version>\<language>\auth\
```

Wenn Sie versuchen, eine Verbindung zu SQL Server-Datenbank unter Verwendung der Windows-Authentifizierung herzustellen, ist es möglich, dass der folgende Fehler auftritt: Dieser Treiber ist nicht für die integrierte Authentifizierung konfiguriert. Dieses Problem kann gelöst werden, indem Sie die folgenden Aktionen ausführen:

- Deklarieren Sie zwei Variablen, die auf den installierten Pfad Ihres JDBC verweisen:

```
variable name: SQLJDBC_HOME; variable value: D:\lib\JDBC4.1\enu(wo Ihre  
sqljdbc4.jar existiert);
```

```
variable name: SQLJDBC_AUTH_HOME; variable value: D\lib\JDBC4.1\enu  
\auth\x86(wenn Sie ein 32-Bit-Betriebssystem verwenden) oderD\lib\JDBC4.1\enu\auth  
\x64(wenn Sie ein 64-Bit-Betriebssystem verwenden). Hier ist deinsqljdbc_auth.dllbefindet  
sich.
```

- Kopieren `sqljdbc_auth.dll` in den Ordner, in dem Ihr JDK/JRE läuft. Sie können in den lib-Ordner, in den bin-Ordner usw. kopieren. Beispielsweise können Sie in den folgenden Ordner kopieren.

```
[JDK_INSTALLED_PATH]\bin;  
[JDK_INSTALLED_PATH]\jre\bin;  
[JDK_INSTALLED_PATH]\jre\lib;  
[JDK_INSTALLED_PATH]\lib;
```

- Stellen Sie sicher, dass Sie in Ihrem JDBC-Bibliotheksordner nur die Datei `SQLJDBC4.jar` haben. Entfernen Sie alle anderen `sqljdbc*.jar`-Dateien aus diesem Ordner (oder kopieren Sie sie in einen

anderen Ordner). Wenn Sie den Treiber als Teil Ihres Programms hinzufügen, stellen Sie sicher, dass Sie nur SQLJDBC4.jar als zu verwendenden Treiber hinzufügen.

- Kopieren Sie die sqljdbc_auth.dll-Datei in den Ordner mit Ihrer Anwendung.

Note

Wenn Sie eine 32-Bit-JVM (Java Virtual Machine) ausführen, verwenden Sie die sqljdbc_auth.dll-Datei im Ordner x86, auch wenn das Betriebssystem die x64-Version ist. Wenn Sie eine 64-Bit-JVM auf einem x64-Prozessor ausführen, verwenden Sie die sqljdbc_auth.dll-Datei im Ordner x64.

Wenn Sie eine Verbindung zu einer SQL Server-Datenbank herstellen, können Sie entweder Windows Authentication (Windows-Authentifizierung) oder SQL Server Authentication (SQL Server-Authentifizierung) für die Option Authentication (Authentifizierung) verwenden.

Verbindung mit SQL Server als -Quelle herstellen.

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Microsoft SQL Server-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Microsoft SQL Server-Quelldatenbank her

1. In der AWS Schema Conversion Tool, wähle Quelle hinzufügen.
2. Wähle Microsoft SQL Server, wählen Sie dann Als Nächstes.

Das Quelle hinzufügen Zeigen Sie ein Dialogfeld an.

3. Für Name der Verbindung, geben Sie einen Namen für die -Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie die Anmeldeinformationen der Datenbank aus AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um Datenbankanmeldedaten von Secrets Manager zu verwenden:
 1. Für AWS-Geheimnis, wähle die -Prozedur.
 2. Wähle Bevölkerung um alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager automatisch auszufüllen.

Hinweise zur Verwendung der Datenbank-Anmeldeinformationen aus dem Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen zur Microsoft SQL Server-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	<p>Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Quelldatenbankservers ein.</p> <p>Sie können mithilfe eines IPv6-Adressprotokolls auf der SQL Server-Datenbank verwenden. Stellen Sie dazu sicher, dass Sie die IP-Adresse in eckigen Klammern eingeben, wie im folgenden Beispiel gezeigt.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"> <code>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</code> </div>
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Instance name	Geben Sie den Instance-Namen für die SQL-Serverdatenbank ein. Um den Instance-Namen abzurufen, senden Sie die Abfrage <code>SELECT @@servername;</code> an Ihre SQL-Serverdatenbank.
Authentifizierung	Wählen Sie die -Prozedur aus. Windows-Authentifizierung und SQL Server-Authentifizierung.

Parameter	Action
User name und Password	<p>Geben Sie die Datenbank-Anmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver an.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich dafür entscheiden, eine Verbindung zu Ihrer Datenbank in einem Projekt herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, AWS SCT speichert das Passwort standardmäßig nicht. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>
Use SSL	<p>Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung zu Ihrer Datenbank zu verwenden. Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der SSL-Registerkarte:</p> <ul style="list-style-type: none">• Vertrauen Sie dem Serverzertifikat: Wählen Sie diese Option, um dem Serverzertifikat zu vertrauen.• Vertrauenswürdiger Speicher: Der Speicherort eines Vertrauensspeichers, der Zertifikate enthält. Damit dieser Speicherort in der Globale Einstellungen-Abschnitt, stellen Sie sicher, dass Sie ihn hinzufügen.
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankpasswort speichern und schnell auf die Datenbank zugreifen, ohne dass Sie das Passwort eingeben müssen.</p>

Parameter	Action
Sql Server Driver Path	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projekteinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>
Bibliothek für die Windows-Authentifizierung	<p>Geben Sie den Pfad zur <code>sqljdbc_auth.dll</code> Datei. Standardmäßig ist diese Datei am folgenden Speicherort installiert:</p> <p><i><installation directory of the JDBC driver>sqljdbc_<version> \<language> \auth\</i></p>

5. Wählen Sie `test` aus, um die Verbindung zu Ihrer Quelldatenbank zu überprüfen.
6. Wählen Sie `connect` aus, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Umwandeln von SQL Server in MySQL

Um Microsoft SQL Server-Datenbankfunktionen in Ihrem konvertierten MySQL-Code zu emulieren, verwenden Sie das Erweiterungspaket SQL Server to MySQL in AWS SCT. Weitere Informationen über die `-Prozedur` finden Sie unter [Verwenden von AWS SCT Erweiterungspaketen](#).

Themen

- [Rechte für MySQL als Zieldatenbank](#)
- [Einstellungen für die Konvertierung von SQL Server nach MySQL](#)
- [Überlegungen zur Migration](#)

Rechte für MySQL als Zieldatenbank

Die für MySQL als Ziel erforderlichen Rechte lauten wie folgt:

- ERSTELLEN AM * . *
- SPÄTER . *
- VORBEIKOMMEN * . *
- INDEX AUF * . *
- VERWEISE AUF * . *
- SELECT ON *.*
- ANSICHT ERSTELLEN AUF * . *
- SHOW VIEW ON *.*
- TRIGGER AUF * . *
- ROUTINE ERSTELLEN AUF * . *
- ROUTINE ÄNDERN AUF * . *
- AUSFÜHREN AM * . *
- EINFÜGEN, AUF AWS_SQLSERVER_EXT AKTUALISIEREN. *
- EINFÜGEN, AKTUALISIEREN, LÖSCHEN AUF AWS_SQLSERVER_EXT_DATA. *
- ERSTELLEN SIE TEMPORÄRE TABELLEN AUF AWS_SQLSERVER_EXT_DATA. *

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Rechte zu gewähren.

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';
GRANT CREATE ON *.* TO 'user_name';
GRANT ALTER ON *.* TO 'user_name';
GRANT DROP ON *.* TO 'user_name';
GRANT INDEX ON *.* TO 'user_name';
GRANT REFERENCES ON *.* TO 'user_name';
GRANT SELECT ON *.* TO 'user_name';
GRANT CREATE VIEW ON *.* TO 'user_name';
GRANT SHOW VIEW ON *.* TO 'user_name';
GRANT TRIGGER ON *.* TO 'user_name';
GRANT CREATE ROUTINE ON *.* TO 'user_name';
GRANT ALTER ROUTINE ON *.* TO 'user_name';
GRANT EXECUTE ON *.* TO 'user_name';
GRANT INSERT, UPDATE ON AWS_SQLSERVER_EXT.* TO 'user_name';
GRANT INSERT, UPDATE, DELETE ON AWS_SQLSERVER_EXT_DATA.* TO 'user_name';
GRANT CREATE TEMPORARY TABLES ON AWS_SQLSERVER_EXT_DATA.* TO 'user_name';
```

Im vorherigen Beispiel ersetzen `user_name` mit dem Namen Ihres Benutzers. Dann ersetze `dein_Passwort` mit einem sicheren Passwort.

Wenn Sie eine MySQL-Datenbank Version 5.7 oder niedriger als Ziel verwenden, führen Sie den folgenden Befehl aus. Für MySQL-Datenbanken der Version 8.0 und höher ist dieser Befehl veraltet.

```
GRANT SELECT ON mysql.proc TO 'user_name';
```

Um Amazon RDS for MySQL oder Aurora MySQL als Ziel zu verwenden, legen Sie den `lower_case_table_names` Parameter auf 1. Dieser Wert bedeutet, dass der MySQL-Server Bezeichner von Objektnamen wie Tabellen, Indizes, Triggern und Datenbanken ohne Berücksichtigung der Groß- und Kleinschreibung behandelt. Wenn Sie die binäre Protokollierung in Ihrer Zielinstanz aktiviert haben, legen Sie den `log_bin_trust_function_creators` Parameter auf 1. In diesem Fall müssen Sie die `-Prozedur` nicht verwenden. `DETERMINISTIC,READS SQL DATA` oder `NO SQL` Eigenschaften, um gespeicherte Funktionen zu erstellen. Um diese Parameter zu konfigurieren, erstellen Sie eine neue DB-Parametergruppe oder ändern Sie eine bestehende DB-Parametergruppe.

Einstellungen für die Konvertierung von SQL Server nach MySQL

Um die Einstellungen für die Konvertierung von SQL Server nach MySQL zu bearbeiten, in AWS SCT wählen Einstellungen, und wählen Sie dann Einstellungen für die Konvertierung. Wählen Sie aus der oberen Liste SQL-Server, und wählen Sie dann SQL-Server — MySQL. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von SQL Server zu MySQL an.

Einstellungen für die Konvertierung von SQL Server zu MySQL in AWS SCT beinhaltet -Optionen für die `-Prozedur`.

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Für Fügen Sie dem konvertierten Code Kommentare für die Aktionselemente mit dem ausgewählten Schweregrad und höher hinzu, wählen Sie den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Damit Ihre SQL Server-Quelldatenbank die Ausgabe von speichern kannEXECin einer Tabelle.AWS SCTerstellt temporäre Tabellen und ein zusätzliches Verfahren, um diese Funktion zu emulieren. Um diese Emulation zu verwenden, wählen SieErstellen Sie zusätzliche Routinen für den Umgang mit offenen Datensätzen.

Überlegungen zur Migration

Beachten Sie bei der Migration eines SQL Server-Schemas zu MySQL die folgenden Punkte:

- MySQL unterstützt das nichtMERGEAnweisung. JedochAWS SCTkann das emulierenMERGEAnweisung während der Konvertierung mit demINSERT ON DUPLICATE KEYKlausel und dieUPDATE FROM and DELETE FROMAnweisungen.

Für eine korrekte Emulation mitINSERT ON DUPLICATE KEY, stellen Sie sicher, dass eine eindeutige Einschränkung oder ein eindeutiger Primärschlüssel in der MySQL-Zieldatenbank vorhanden ist.

- Sie können einGOTOAnweisung und ein Label, um die Reihenfolge zu ändern, in der Anweisungen ausgeführt werden. Alle Transact-SQL-Anweisungen, die auf eineGOTODie Anweisung wird übersprungen, und die Verarbeitung wird am Label fortgesetzt. Sie könnenGOTOAnweisungen und Labels an einer beliebigen Stelle innerhalb einer Prozedur, eines Batches oder eines Anweisungsblocks. Sie können auch nistenGOTOAnweisungen.

MySQL verwendet nichtGOTOAnweisungen. WannAWS SCTkonvertiert Code, der ein enthältGOTOAnweisung, es konvertiert die Anweisung so, dass sie eine verwendetBEGIN... ENDoderLOOP...END LOOPAnweisung. Sie können Beispiele dafür findenAWS SCTkonvertiertGOTOAnweisungen in der folgenden Tabelle.

SQL Server-Anweisung	MySQL-Anweisung
<pre>BEGIN statement1; GOTO label1; statement2; label1: Statement3;</pre>	<pre>BEGIN label1: BEGIN statement1; LEAVE label1; statement2; </pre>

SQL Server-Anweisung	MySQL-Anweisung
<pre>.... END</pre>	<pre>END; Statement3; END</pre>
<pre>BEGIN statement1; label1: statement2; GOTO label1; statement3; statement4; END</pre>	<pre>BEGIN statement1; label1: LOOP statement2; ITERATE label1; LEAVE label1; END LOOP; statement3; statement4; END</pre>
<pre>BEGIN statement1; label1: statement2; statement3; statement4; END</pre>	<pre>BEGIN statement1; label1: BEGIN statement2; statement3; statement4; END; END</pre>

- MySQL unterstützt keine Tabellenwertfunktionen mit mehreren Anweisungen. AWS SCT simuliert Tabellenwertfunktionen während einer Konvertierung, indem temporäre Tabellen erstellt und Anweisungen so umgeschrieben werden, dass sie diese temporären Tabellen verwenden.

Umwandeln von SQL Server in PostgreSQL

Sie können das Erweiterungspaket SQL Server to PostgreSQL in verwenden AWS SCT. Dieses Erweiterungspaket emuliert SQL Server-Datenbankfunktionen im konvertierten PostgreSQL-Code. Verwenden Sie das Erweiterungspaket SQL Server to PostgreSQL, um SQL Server Agent und SQL Server Database Mail zu emulieren. Weitere Informationen über die -Prozedur finden Sie unter [Verwenden von AWS SCT Erweiterungspaketen](#).

Themen

- [Rechte für PostgreSQL als Zieldatenbank](#)
- [Einstellungen für die Konvertierung von SQL Server nach PostgreSQL](#)
- [Konvertieren von SQL Server-Partitionen in PostgreSQL Version 10-Partitionen](#)
- [Überlegungen zur Migration](#)
- [Mit einem AWS SCT Erweiterungspaket zur Emulation des SQL Server-Agents in PostgreSQL](#)
- [Mit einem AWS SCT Erweiterungspaket zur Emulation von SQL Server-Datenbank-Mail in PostgreSQL](#)

Rechte für PostgreSQL als Zieldatenbank

Um PostgreSQL als Ziel zu verwenden, AWS SCT benötigt CREATE ON DATABASE Privileg. Stellen Sie sicher, dass Sie dieses Recht für jede PostgreSQL-Zieldatenbank gewähren.

Um die konvertierten öffentlichen Synonyme zu verwenden, ändern Sie den Standardsuchpfad der Datenbank in "\$user", public_synonyms, public.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die entsprechenden Rechte zu gewähren.

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';  
GRANT CREATE ON DATABASE db_name TO user_name;  
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

Im vorherigen Beispiel ersetzen `user_name` mit dem Namen Ihres Benutzers. Dann ersetze `db_name` mit der -Prozedur -Prozedur. Schließlich ersetzend `dein_Passwort` mit einem sicheren Passwort.

In PostgreSQL nur der Schemabesitzer oder ein `superuser` kann ein Schema löschen. Der Besitzer kann ein Schema und alle Objekte, die dieses Schema enthält, löschen, auch wenn der Eigentümer des Schemas einige seiner Objekte nicht besitzt.

Wenn Sie verschiedene Benutzer verwenden, um verschiedene Schemas zu konvertieren und auf Ihre Zieldatenbank anzuwenden, kann in folgenden Fällen eine Fehlermeldung angezeigt werden AWS SCT kann ein Schema nicht löschen. Um die -Prozedur zu vermeiden, verwenden Sie die -Prozedur `superuser` Rolle.

Einstellungen für die Konvertierung von SQL Server nach PostgreSQL

Um die Einstellungen für die Konvertierung von SQL Server nach PostgreSQL zu bearbeiten, wählen Sie `Einstellungen` und dann `Einstellungen für die Konvertierung`. Wählen Sie aus der oberen Liste `SQL-Server` und dann `SQL-Server — PostgreSQL`. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von SQL Server zu PostgreSQL an.

Einstellungen für die Konvertierung von SQL Server nach PostgreSQL in AWS SCT beinhaltet - Optionen für die -Prozedur.

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Für Fügen Sie dem konvertierten Code Kommentare für die Aktionselemente mit dem ausgewählten Schweregrad und höher hinzu, wählen Sie den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie `Nur Fehler`. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie `Alle Nachrichten`.

- Um die Verwendung von Indizes mit demselben Namen in verschiedenen Tabellen in SQL Server zu ermöglichen.

In PostgreSQL müssen alle Indexnamen, die Sie im Schema verwenden, eindeutig sein. Um sicherzustellen, dass AWS SCT generiert eindeutige Namen für alle Ihre Indizes, wählen Sie `Generieren Sie eindeutige Namen für Indizes`.

- Um SQL Server-Prozeduren in PostgreSQL-Funktionen zu konvertieren.

PostgreSQL Version 10 und früher unterstützt keine Prozeduren. Für Kunden, die mit der Verwendung von Verfahren in PostgreSQL nicht vertraut sind, AWS SCT kann Prozeduren in Funktionen umwandeln. Wählen Sie End und dann Prozeduren in Funktionen umwandeln.

- Um die Ausgabe von zu emulieren EXEC in einer Tabelle.

Ihre SQL Server-Quelldatenbank kann die Ausgabe von speichern EXEC in einer Tabelle. AWS SCT erstellt temporäre Tabellen und ein zusätzliches Verfahren, um diese Funktion zu emulieren. Um diese Emulation zu verwenden, wählen Sie Erstellen Sie zusätzliche Routinen für den Umgang mit offenen Datensätzen.

- Um die Vorlage zu definieren, die für die Schemanamen im konvertierten Code verwendet werden soll. Für -Vorlage für die -Prozedur., wählen Sie eine der folgenden -Prozedur:
 - <source_db>— Verwendet den SQL Server-Datenbanknamen als Schemanamen in PostgreSQL.
 - <source_schema>— Verwendet den SQL Server-Schemanamen als Schemanamen in PostgreSQL.
 - <source_db>_ <schema>— Verwendet eine Kombination aus der SQL Server-Datenbank und Schemanamen als Schemanamen in PostgreSQL.
- Um die Groß- und Kleinschreibung Ihrer Quellobjektnamen beizubehalten.

Um zu verhindern, dass Objektnamen in Kleinbuchstaben umgewandelt werden, wählen Sie Vermeiden Sie bei Operationen, bei denen Groß- und Kleinschreibung beachtet wird. Diese Option gilt nur, wenn Sie die Option zur Berücksichtigung von Groß- und Kleinschreibung in Ihrer Zieldatenbank aktivieren.

- Um die Parameternamen aus Ihrer Quelldatenbank beizubehalten.

Um den Namen der Parameter im konvertierten Code doppelte Anführungszeichen hinzuzufügen, wählen Sie Behalten Sie die ursprünglichen Parameternamen bei.

Konvertieren von SQL Server-Partitionen in PostgreSQL Version 10-Partitionen

Beachten Sie Folgendes, wenn Sie eine Microsoft SQL Server-Datenbank in Amazon Aurora PostgreSQL-Compatible Edition (Aurora PostgreSQL) oder Amazon Relational Database Service for PostgreSQL (Amazon RDS for PostgreSQL) konvertieren.

In SQL Server können Sie Partitionen mit Partitionsfunktionen erstellen. Bei der Umwandlung einer partitionierten SQL Server-Tabelle in eine partitionierte PostgreSQL Version 10-Tabelle sind die folgenden potenziellen Probleme zu beachten:

- SQL Server ermöglicht es Ihnen, eine Tabelle mit einer Spalte ohne NOT NULL-Einschränkung zu partitionieren. In diesem Fall werden alle NULL-Werte in die ganz links befindliche Partition verschoben. PostgreSQL unterstützt keine NULL-Werte für RANGE-Partitionierungen.
- SQL Server ermöglicht Ihnen die Erstellung primärer und eindeutiger Schlüssel für partitionierte Tabellen. Sie erstellen für PostgreSQL primäre oder eindeutige Schlüssel direkt für jede Partition. Daher muss bei einer Migration in PostgreSQL die PRIMARY KEY- oder UNIQUE KEY-Einschränkung von ihrer übergeordneten Tabelle entfernt werden. Die resultierenden Schlüsselnamen haben das folgende Format `<original_key_name>_<partition_number>`.
- SQL Server ermöglicht Ihnen die Erstellung von Einschränkungen für Fremdschlüsselverweise aus und in partitionierte Tabellen. PostgreSQL unterstützt keine Fremdschlüssel, die auf partitionierte Tabellen verweisen. PostgreSQL unterstützt auch keine Fremdschlüsselverweise aus einer partitionierten Tabelle in eine andere Tabelle.
- SQL Server ermöglicht Ihnen die Erstellung von Indizes für partitionierte Tabellen. Für PostgreSQL sollte für jede Partition direkt ein Index erstellt werden. Folglich müssen Indizes bei der Migration in PostgreSQL aus den ihnen übergeordneten Tabellen entfernt werden. Die sich ergebenden Indexnamen weisen das Format `<original_index_name>_<partition_number>` auf.
- PostgreSQL unterstützt keine partitionierten Indizes.

Überlegungen zur Migration

Einige Dinge, die Sie bei der Migration eines SQL Server-Schemas nach PostgreSQL beachten sollten:

- In PostgreSQL müssen alle Objektnamen in einem Schema eindeutig sein. Dies gilt auch für Indizes. Indexnamen müssen im Schema der Basistabelle eindeutig sein. In SQL Server kann ein Indexname für verschiedene Tabellen gleich sein.

Um die Einzigartigkeit von Indexnamen sicherzustellen, AWS SCT bietet Ihnen die Möglichkeit, eindeutige Indexnamen zu generieren, wenn Ihre Indexnamen nicht eindeutig sind. Wählen Sie hierzu die Option `Generate unique index names` (Eindeutige Indexnamen generieren) in den Projekteigenschaften. Diese Funktion ist standardmäßig aktiviert. Wenn diese Option aktiviert ist, werden eindeutige Indexnamen im Format `IX_table_name_index_name` erstellt. Wenn diese Option deaktiviert ist, werden Indexnamen nicht geändert.

- Mithilfe einer GOTO-Anweisung und einer Bezeichnung lässt sich die Reihenfolge ändern, in der Anweisungen ausgeführt werden. Alle Transact-SQL-Anweisungen, die auf eine GOTO-Anweisung folgen, werden übersprungen, und die Verarbeitung wird ab der Bezeichnung fortgesetzt. GOTO-Anweisungen und Bezeichnungen können an beliebiger Stelle innerhalb eines Verfahrens, Stapels oder Anweisungsblocks verwendet werden. GOTO-Anweisungen können auch verschachtelt werden.

PostgreSQL verwendet keine GOTO-Anweisungen. Wenn AWS SCT konvertiert Code, der eine GOTO-Anweisung enthält, und konvertiert die Anweisung so, dass sie eine BEGIN... END- oder LOOP... END LOOP-Anweisung verwendet. Sie können Beispiele dafür finden AWS SCT konvertiert GOTO-Anweisungen in der folgenden Tabelle.

SQL Server GOTO-Anweisungen und die konvertierten PostgreSQL-Anweisungen

SQL Server-Anweisung	PostgreSQL-Anweisung
<pre>BEGIN statement1; GOTO label1; statement2; label1: Statement3; END</pre>	<pre>BEGIN label1: BEGIN statement1; EXIT label1; statement2; END; Statement3; END</pre>

SQL Server-Anweisung	PostgreSQL-Anweisung
<pre> BEGIN statement1; label1: statement2; GOTO label1; statement3; statement4; END </pre>	<pre> BEGIN statement1; label1: LOOP statement2; CONTINUE label1; EXIT label1; END LOOP; statement3; statement4; END </pre>
<pre> BEGIN statement1; label1: statement2; statement3; statement4; END </pre>	<pre> BEGIN statement1; label1: BEGIN statement2; statement3; statement4; END; END </pre>

- PostgreSQL unterstützt keine MERGE-Anweisung. AWS SCT emuliert die -Prozedur auf die -Prozedur.
 - Durch INSERT ON CONFLICT-Konstruktion.
 - Durch die Verwendung der UPDATE FROM DML-Anweisung, wie z. B. MERGE ohne WHEN NOT MATCHED-Klausel.

- Durch die Verwendung von CURSOR, z. B. mit einer MERGE-Anweisung mit DELETE-Klausel, oder durch die Verwendung einer komplexen MERGE ON-Bedingungsanweisung.
- AWS SCT kann Datenbank-Trigger zum Objektbaum hinzufügen, wenn Amazon RDS das Ziel ist.
- AWS SCT kann Trigger auf Serverebene zum Objektbaum hinzufügen, wenn Amazon RDS das Ziel ist.
- SQL Server erstellt und verwaltet automatisch `deleted` und `inserted` Tische. Sie können diese temporären, speicherresidenten Tabellen verwenden, um die Auswirkungen bestimmter Datenänderungen zu testen und Bedingungen für DML-Triggeraktionen festzulegen. AWS SCT kann die Verwendung dieser Tabellen innerhalb von DML-Triggeranweisungen konvertieren.
- AWS SCT kann verknüpfte Server zum Objektbaum hinzufügen, wenn Amazon RDS das Ziel ist.
- Bei der Migration von Microsoft SQL Server zu PostgreSQL wird die integrierte `SUSER_SNAME`-Funktion wie folgt umgewandelt:
 - `SUSER_SNAME` – Gibt den Anmeldenamen zurück, der einer Sicherheits-ID (SID) zugeordnet ist.
 - `SUSER_SNAME(<server_user_sid>)` – Nicht unterstützt.
 - `SUSER_SNAME() CURRENT_USER` – Gibt den Benutzernamen des aktuellen Ausführungskontexts zurück.
 - `SUSER_SNAME(NULL)` – Gibt NULL zurück.
- Die Umwandlung von Tabellenwertfunktionen wird unterstützt. Tabellenwertfunktionen geben eine Tabelle zurück und können in einer Abfrage die Stelle einer Tabelle einnehmen.
- `PATINDEX` gibt die Startposition des ersten Auftretens eines Musters in einem bestimmten Ausdruck zurück. Dabei werden alle gültigen Text und Zeichendatentypen unterstützt. Nullen werden zurückgegeben, wenn das Muster nicht gefunden wird. Bei der Konvertierung von SQL Server zu Amazon RDS for PostgreSQL wird `aws_sct<pattern character><expression character varying>` ersetzt den Anwendungscode, der `PATINDEX` verwendet, durch `aws_sqlserver_ext.patindex (,)`.
- In SQL Server ist ein benutzerdefinierter Tabellentyp ein Typ, der die Definition einer Tabellenstruktur darstellt. Sie verwenden einen benutzerdefinierten Tabellentyp, um table-value-Parameter für gespeicherte Prozeduren oder Funktionen zu deklarieren. Sie können auch einen benutzerdefinierten Tabellentyp verwenden, um Tabellenvariablen zu deklarieren, die Sie in einem Batch oder im Hauptteil einer gespeicherten Prozedur oder Funktion verwenden möchten. AWS SCT emuliert diesen Typ in PostgreSQL, indem eine temporäre Tabelle erstellt wurde.

Bei der Konvertierung von SQL Server nach PostgreSQL AWS SCT konvertiert SQL Server-Systemobjekte in erkennbare Objekte in PostgreSQL. Die folgende Tabelle zeigt, wie die Systemobjekte umgewandelt werden.

Anwendungsfälle für MS SQL Server	PostgreSQL-Substitution
SYS.SCHEMAS	AWS_SQLSERVER_EXT.SYS_SCHEMAS
SYS.TABLES	AWS_SQLSERVER_EXT.SYS_TABLES
SYS.VIEWS	AWS_SQLSERVER_EXT.SYS_VIEWS
SYS.ALL_VIEWS	AWS_SQLSERVER_EXT.SYS_ALL_VIEWS
SYS.TYPES	AWS_SQLSERVER_EXT.SYS_TYPES
SYS.COLUMNS	AWS_SQLSERVER_EXT.SYS_COLUMNS
SYS.ALL_COLUMNS	AWS_SQLSERVER_EXT.SYS_ALL_COLUMNS
SYS.FOREIGN_KEYS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEYS
SYS.SYSFOREIGNKEYS	AWS_SQLSERVER_EXT.SYS_SYSFOREIGNKEYS
SYS.FOREIGN_KEY_COLUMNS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEY_COLUMNS
SYS.KEY_CONSTRAINTS	AWS_SQLSERVER_EXT.SYS_KEY_CONSTRAINTS
SYS.IDENTITY_COLUMNS	AWS_SQLSERVER_EXT.SYS_IDENTITY_COLUMNS
SYS.PROCEDURES	AWS_SQLSERVER_EXT.SYS_PROCEDURES
SYS.INDEXES	AWS_SQLSERVER_EXT.SYS_INDEXES
SYS.SYSINDEXES	AWS_SQLSERVER_EXT.SYS_SYSINDEXES
SYS.OBJECTS	AWS_SQLSERVER_EXT.SYS_OBJECTS

Anwendungsfälle für MS SQL Server	PostgreSQL-Substitution
SYS.ALL_OBJECTS	AWS_SQLSERVER_EXT.SYS_ALL_OBJECTS
SYS.SYSOBJECTS	AWS_SQLSERVER_EXT.SYS_SYSOBJECTS
SYS.SQL_MODULES	AWS_SQLSERVER_EXT.SYS_SQL_MODULES
SYS.DATABASES	AWS_SQLSERVER_EXT.SYS_DATABASES
INFORMATION_SCHEMA .SCHEMATA	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_SCHEMATA
INFORMATION_SCHEMA .VIEWS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_VIEWS
INFORMATION_SCHEMA .TABLES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLES
INFORMATION_SCHEMA .COLUMNS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_COLUMNS
INFORMATION_SCHEMA .CHECK_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CHECK_CONSTRAINTS
INFORMATION_SCHEMA .REFERENTIAL_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_REFERENTIAL_CONSTRAINTS
INFORMATION_SCHEMA .TABLE_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLE_CONSTRAINTS
INFORMATION_SCHEMA .KEY_COLUMN_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_KEY_COLUMN_USAGE
INFORMATION_SCHEMA .CONSTRAINT_TABLE_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAINT_TABLE_USAGE

Anwendungsfälle für MS SQL Server	PostgreSQL-Substitution
INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAINT_COLUMN_USAGE
INFORMATION_SCHEMA.ROUTINES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_ROUTINES
SYS.SYSPROCESSES	AWS_SQLSERVER_EXT.SYS_SYSPROCESSES
sys.system_objects	AWS_SQLSERVER_EXT.SYS_SYSTEM_OBJECTS

Mit einem AWS SCT Erweiterungspaket zur Emulation des SQL Server-Agents in PostgreSQL

Der SQL Server Agent ist ein Microsoft Windows-Dienst, der SQL Server-Jobs ausführt. Der SQL Server-Agent führt Jobs nach einem Zeitplan, als Reaktion auf ein bestimmtes Ereignis oder auf der Anforderung aus. Weitere Informationen über die SQL Server-Agent finden Sie unter [Technische Dokumentation von Microsoft](#).

PostgreSQL hat keine -Prozedur für die SQL Server-Agentin. Um die Funktionen des SQL Server-Agents zu emulieren, AWS SCT erstellt ein Erweiterungspaket. Dieses Erweiterungspaket verwendet AWS Lambda und Amazon CloudWatch. AWS Lambda implementiert die Schnittstelle, mit der Sie Zeitpläne verwalten und Jobs ausführen. Amazon CloudWatch behält die Zeitplanregeln bei.

AWS Lambda und Amazon CloudWatch verwenden Sie einen JSON-Parameter, um zu interagieren. Dieser JSON-Parameter hat die -Prozedur.

```
{
  "mode": mode,
  "parameters": {
    list of parameters
  },
  "callback": procedure name
}
```

Im vorherigen Beispiel *mode* ist der Typ der Aufgabe und *list of parameters* ist eine -Prozedur, die von der -Prozedur abhängt. Außerdem *procedure name* ist der Name der Prozedur, die nach Abschluss der Aufgabe ausgeführt wird.

AWS SCT verwendet eine Lambda-Funktion zur Steuerung und Ausführung von Jobs. Das CloudWatch Die Regel startet die Ausführung des Jobs und stellt die notwendigen Informationen bereit, um den Job zu starten. Wenn der CloudWatch Regel löst aus, sie startet die Lambda-Funktion unter Verwendung der Parameter aus der Regel.

Verwenden Sie das folgende Format, um einen einfachen Job zu erstellen, der eine Prozedur aufruft.

```
{
  "mode": "run_job",
  "parameters": {
    "vendor": "mysql",
    "cmd": "lambda_db.nightly_job"
  }
}
```

Um einen Job mit mehreren -Schritten zu erstellen, verwenden Sie die -Prozedur.

```
{
  "mode": "run_job",
  "parameters": {
    "job_name": "Job1",
    "enabled": "true",
    "start_step_id": 1,
    "notify_level_email": [0|1|2|3],
    "notify_email": email,
    "delete_level": [0|1|2|3],
    "job_callback": "ProcCallBackJob(job_name, code, message)",
    "step_callback": "ProcCallBackStep(job_name, step_id, code, message)"
  },
  "steps": [
    {
      "id": 1,
      "cmd": "ProcStep1",
      "cmdexec_success_code": 0,
      "on_success_action": [2|3|4],
      "on_success_step_id": 1,
      "on_fail_action": 0,
      "on_fail_step_id": 0,
    }
  ]
}
```

```
        "retry_attempts": number,  
        "retry_interval": number  
    },  
    {  
        "id":2,  
        "cmd": "ProcStep2",  
        "cmdexec_success_code": 0,  
        "on_success_action": [1|2|3|4],  
        "on_success_step_id": 0,  
        "on_fail_action": 0,  
        "on_fail_step_id": 0,  
        "retry_attempts": number,  
        "retry_interval": number  
    },  
    ...  
]  
}
```

Um das Verhalten des SQL Server-Agents in PostgreSQL zu emulieren, AWS SCT Das Erweiterungspaket erstellt auch die folgenden Tabellen und Prozeduren.

Tabellen, die den SQL Server-Agent in PostgreSQL emulieren

Um den SQL Server-Agent zu emulieren, verwendet das Erweiterungspaket die folgenden Tabellen:

sysjobs

Speichert die Informationen über die -Prozedur.

sysjobsteps

Speichert die Informationen über die -Prozedur.

Sysschedules

Speichert die Informationen über die -Prozedur.

sysjobschedules

Speichert die Zeitplaninformationen für einzelne Jobs.

sysjobhistory

Speichert die -Prozedur.

Verfahren, die den SQL Server-Agent in PostgreSQL emulieren

Um den SQL Server-Agent zu emulieren, verwendet das Erweiterungspaket die folgenden Verfahren:

sp_add_job

Fügt einen neuen Job hinzu.

sp_add_jobstep

Fügt einem Job einen Schritt hinzu.

sp_add_schedule

Erstellt eine neue Zeitplanregel in Amazon CloudWatch. Sie können diesen Zeitplan für eine beliebige Anzahl von Jobs verwenden.

sp_attach_schedule

Legt einen Zeitplan für den ausgewählten Job fest.

sp_add_jobschedule

Erstellt eine Zeitplanregel für einen Job in Amazon CloudWatch und legt das Ziel für diese Regel fest.

sp_update_job

Aktualisiert die Attribute des zuvor erstellten Jobs.

sp_update_jobstep

Aktualisiert die Attribute des Schritts in einem Job.

sp_update_schedule

Aktualisiert die Attribute einer Zeitplanregel in Amazon CloudWatch.

sp_update_jobschedule

Aktualisiert die -Prozedur für die -Prozedur.

sp_delete_job

Löscht einen Job aus.

sp_delete_jobstep

Löscht einen Job-Schritt aus einem Job.

sp_delete_schedule

Löscht einen Zeitplan.

sp_delete_jobschedule

Löscht die Zeitplanregel für den angegebenen Job aus Amazon CloudWatch.

sp_detach_schedule

Löscht eine Verbindung zwischen einem Zeitplan und einem Job.

get_jobs, update_job

Interne Verfahren, die interagieren mit AWS Elastic Beanstalk.

sp_verify_job_date, sp_verify_job_time, sp_verify_job, sp_verify_jobstep, sp_verify_schedule, sp_verify_job_identifiers, sp_verify_schedule_identifiers

Interne Verfahren zur Überprüfung der Einstellungen.

Syntax für Prozeduren, die den SQL Server-Agent in PostgreSQL emulieren

Das `aws_sqlserver_ext.sp_add_job` Verfahren im Erweiterungspaket emuliert das `msdb.dbo.sp_add_job` Verfahren. Weitere Informationen über die SQL Server-Agent-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_job_name varchar,  
par_enabled smallint = 1,  
par_description varchar = NULL::character varying,  
par_start_step_id integer = 1,  
par_category_name varchar = NULL::character varying,  
par_category_id integer = NULL::integer,  
par_owner_login_name varchar = NULL::character varying,  
par_notify_level_eventlog integer = 2,  
par_notify_level_email integer = 0,  
par_notify_level_netsend integer = 0,  
par_notify_level_page integer = 0,  
par_notify_email_operator_name varchar = NULL::character varying,  
par_notify_netsend_operator_name varchar = NULL::character varying,  
par_notify_page_operator_name varchar = NULL::character varying,  
par_delete_level integer = 0,  
inout par_job_id integer = NULL::integer,  
par_originating_server varchar = NULL::character varying,  
out returncode integer
```

Das `aws_sqlserver_ext.sp_add_jobstep` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sp_add_jobstep` Verfahren. Weitere Informationen über die SQL Server-Agent-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_step_id integer = NULL::integer,  
par_step_name varchar = NULL::character varying,  
par_subsystem varchar = 'TSQL'::bpchar,  
par_command text = NULL::text,  
par_additional_parameters text = NULL::text,  
par_cmexec_success_code integer = 0,  
par_on_success_action smallint = 1,  
par_on_success_step_id integer = 0,  
par_on_fail_action smallint = 2,  
par_on_fail_step_id integer = 0,  
par_server varchar = NULL::character varying,  
par_database_name varchar = NULL::character varying,  
par_database_user_name varchar = NULL::character varying,  
par_retry_attempts integer = 0,  
par_retry_interval integer = 0,  
par_os_run_priority integer = 0,  
par_output_file_name varchar = NULL::character varying,  
par_flags integer = 0,  
par_proxy_id integer = NULL::integer,  
par_proxy_name varchar = NULL::character varying,  
inout par_step_uid char = NULL::bpchar,  
out returncode integer
```

Das `aws_sqlserver_ext.sp_add_schedule` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sp_add_schedule` Verfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_schedule_name varchar,  
par_enabled smallint = 1,  
par_freq_type integer = 0,  
par_freq_interval integer = 0,  
par_freq_subday_type integer = 0,  
par_freq_subday_interval integer = 0,  
par_freq_relative_interval integer = 0,  
par_freq_recurrence_factor integer = 0,  
par_active_start_date integer = NULL::integer,
```



```
par_active_end_date integer = 99991231,  
par_active_start_time integer = 0,  
par_active_end_time integer = 235959,  
par_owner_login_name varchar = NULL::character varying,  
*inout par_schedule_uid char = NULL::bpchar,*  
inout par_schedule_id integer = NULL::integer,  
par_originating_server varchar = NULL::character varying,  
out returncode integer
```

Das `saws_sqlserver_ext.sp_attach_schedule` Verfahren im Erweiterungspaket emuliert das `msmsdb.dbo.sp_attach_schedule` Verfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_schedule_id integer = NULL::integer,  
par_schedule_name varchar = NULL::character varying,  
par_automatic_post smallint = 1,  
out returncode integer
```

Das `saws_sqlserver_ext.sp_add_jobschedule` Verfahren im Erweiterungspaket emuliert das `msmsdb.dbo.sp_add_jobschedule` Verfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_name varchar = NULL::character varying,  
par_enabled smallint = 1,  
par_freq_type integer = 1,  
par_freq_interval integer = 0,  
par_freq_subday_type integer = 0,  
par_freq_subday_interval integer = 0,  
par_freq_relative_interval integer = 0,  
par_freq_recurrence_factor integer = 0,  
par_active_start_date integer = NULL::integer,  
par_active_end_date integer = 99991231,  
par_active_start_time integer = 0,  
par_active_end_time integer = 235959,  
inout par_schedule_id integer = NULL::integer,  
par_automatic_post smallint = 1,  
inout par_schedule_uid char = NULL::bpchar,  
out returncode integer
```

Dasaws_sqlserver_ext.sp_delete_jobDas Verfahren im Erweiterungspaket emuliert dasmsdb.dbo.sp_delete_jobVerfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter[Technische Dokumentation von Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_originating_server varchar = NULL::character varying,  
par_delete_history smallint = 1,  
par_delete_unused_schedule smallint = 1,  
out returncode integer
```

Dasaws_sqlserver_ext.sp_delete_jobstepDas Verfahren im Erweiterungspaket emuliert dasmsdb.dbo.sp_delete_jobstepVerfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter[Technische Dokumentation von Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_step_id integer = NULL::integer,  
out returncode integer
```

Dasaws_sqlserver_ext.sp_delete_jobscheduleDas Verfahren im Erweiterungspaket emuliert dasmsdb.dbo.sp_delete_jobscheduleVerfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter[Technische Dokumentation von Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_name varchar = NULL::character varying,  
par_keep_schedule integer = 0,  
par_automatic_post smallint = 1,  
out returncode integer
```

Dasaws_sqlserver_ext.sp_delete_scheduleDas Verfahren im Erweiterungspaket emuliert dasmsdb.dbo.sp_delete_scheduleVerfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter[Technische Dokumentation von Microsoft](#).

```
par_schedule_id integer = NULL::integer,  
par_schedule_name varchar = NULL::character varying,  
par_force_delete smallint = 0,  
par_automatic_post smallint = 1,  
out returncode integer
```

Das `saws_sqlserver_ext.sp_detach_schedule` Das Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sp_detach_schedule` Verfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_schedule_id integer = NULL::integer,  
par_schedule_name varchar = NULL::character varying,  
par_delete_unused_schedule smallint = 0,  
par_automatic_post smallint = 1,  
out returncode integer
```

Das `saws_sqlserver_ext.sp_update_job` Das Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sp_update_job` Verfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_job_id integer = NULL::integer  
par_job_name varchar = NULL::character varying  
par_new_name varchar = NULL::character varying  
par_enabled smallint = NULL::smallint  
par_description varchar = NULL::character varying  
par_start_step_id integer = NULL::integer  
par_category_name varchar = NULL::character varying  
par_owner_login_name varchar = NULL::character varying  
par_notify_level_eventlog integer = NULL::integer  
par_notify_level_email integer = NULL::integer  
par_notify_level_netsend integer = NULL::integer  
par_notify_level_page integer = NULL::integer  
par_notify_email_operator_name varchar = NULL::character varying  
par_notify_netsend_operator_name varchar = NULL::character varying  
par_notify_page_operator_name varchar = NULL::character varying  
par_delete_level integer = NULL::integer  
par_automatic_post smallint = 1  
out returncode integer
```

Das `saws_sqlserver_ext.sp_update_jobschedule` Das Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sp_update_jobschedule` Verfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_job_id integer = NULL::integer  
par_job_name varchar = NULL::character varying
```

```
par_name varchar = NULL::character varying
par_new_name varchar = NULL::character varying
par_enabled smallint = NULL::smallint
par_freq_type integer = NULL::integer
par_freq_interval integer = NULL::integer
par_freq_subday_type integer = NULL::integer
par_freq_subday_interval integer = NULL::integer
par_freq_relative_interval integer = NULL::integer
par_freq_recurrence_factor integer = NULL::integer
par_active_start_date integer = NULL::integer
par_active_end_date integer = NULL::integer
par_active_start_time integer = NULL::integer
    par_active_end_time integer = NULL::integer
par_automatic_post smallint = 1
out returncode integer
```

Das `aws_sqlserver_ext.sp_update_jobstep` Verfahren im Erweiterungspaket emuliert das `msdb.dbo.sp_update_jobstep` Verfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_job_id integer = NULL::integer
par_job_name varchar = NULL::character varying
par_step_id integer = NULL::integer
par_step_name varchar = NULL::character varying
par_subsystem varchar = NULL::character varying
par_command text = NULL::text
par_additional_parameters text = NULL::text
par_cmdexec_success_code integer = NULL::integer
par_on_success_action smallint = NULL::smallint
par_on_success_step_id integer = NULL::integer
par_on_fail_action smallint = NULL::smallint
par_on_fail_step_id integer = NULL::integer
par_server varchar = NULL::character varying
par_database_name varchar = NULL::character varying
par_database_user_name varchar = NULL::character varying
par_retry_attempts integer = NULL::integer
par_retry_interval integer = NULL::integer
par_os_run_priority integer = NULL::integer
par_output_file_name varchar = NULL::character varying
par_flags integer = NULL::integer
par_proxy_id integer = NULL::integer
par_proxy_name varchar = NULL::character varying
```

```
out returncode integer
```

`Dasaws_sqlserver_ext.sp_update_schedule`Das Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sp_update_schedule`Verfahren. Weitere Hinweise zur SQL Server-Agent-Quellprozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_schedule_id integer = NULL::integer
par_name varchar = NULL::character varying
par_new_name varchar = NULL::character varying
par_enabled smallint = NULL::smallint
par_freq_type integer = NULL::integer
par_freq_interval integer = NULL::integer
par_freq_subday_type integer = NULL::integer
par_freq_subday_interval integer = NULL::integer
par_freq_relative_interval integer = NULL::integer
par_freq_recurrence_factor integer = NULL::integer
par_active_start_date integer = NULL::integer
par_active_end_date integer = NULL::integer
par_active_start_time integer = NULL::integer
par_active_end_time integer = NULL::integer
par_owner_login_name varchar = NULL::character varying
par_automatic_post smallint = 1
out returncode integer
```

Beispiele für die Verwendung von Prozeduren, die den SQL Server-Agent in PostgreSQL emulieren

Um einen neuen Job hinzuzufügen, verwenden Sie `dasaws_sqlserver_ext.sp_add_job`Verfahren wie im Folgenden gezeigt.

```
SELECT * FROM aws_sqlserver_ext.sp_add_job (
    par_job_name := 'test_job',
    par_enabled := 1::smallint,
    par_start_step_id := 1::integer,
    par_category_name := '[Uncategorized (Local)]',
    par_owner_login_name := 'sa');
```

Um einen neuen Job-Schritt hinzuzufügen, verwenden Sie `aws_sqlserver_ext.sp_add_jobstep`Verfahren wie im Folgenden gezeigt.

```
SELECT * FROM aws_sqlserver_ext.sp_add_jobstep (
    par_job_name := 'test_job',
```

```
par_step_id := 1::smallint,  
par_step_name := 'test_job_step1',  
par_subsystem := 'TSQL',  
par_command := 'EXECUTE [dbo].[PROC_TEST_JOB_STEP1];',  
par_server := NULL,  
par_database_name := 'GOLD_TEST_SS');
```

Um einen einfachen Zeitplan hinzuzufügen, verwenden Sie `denaws_sqlserver_ext.sp_add_schedule`Verfahren wie im Folgenden gezeigt.

```
SELECT * FROM aws_sqlserver_ext.sp_add_schedule(  
  par_schedule_name := 'RunOnce',  
  par_freq_type := 1,  
  par_active_start_time := 233000);
```

Um einen Zeitplan für einen Job festzulegen, verwenden Sie `denaws_sqlserver_ext.sp_attach_schedule`Verfahren wie im Folgenden gezeigt.

```
SELECT * FROM aws_sqlserver_ext.sp_attach_schedule (  
  par_job_name := 'test_job',  
  par_schedule_name := 'NightlyJobs');
```

Um einen Zeitplan für einen Job zu erstellen, verwenden Sie `denaws_sqlserver_ext.sp_add_jobschedule`Verfahren wie im Folgenden gezeigt.

```
SELECT * FROM aws_sqlserver_ext.sp_add_jobschedule (  
  par_job_name := 'test_job2',  
  par_name := 'test_schedule2',  
  par_enabled := 1::smallint,  
  par_freq_type := 4,  
  par_freq_interval := 1,  
  par_freq_subday_type := 4,  
  par_freq_subday_interval := 1,  
  par_freq_relative_interval := 0,  
  par_freq_recurrence_factor := 0,  
  par_active_start_date := 20100801,  
  par_active_end_date := 99991231,  
  par_active_start_time := 0,  
  par_active_end_time := 0);
```

Anwendungsbeispiele für die Emulation des SQL Server-Agents in PostgreSQL

Wenn Ihr Quelldatenbankcode den SQL Server-Agent zum Ausführen von Aufträgen verwendet, können Sie das Erweiterungspaket SQL Server to PostgreSQL für verwendenAWS SCTum diesen Code in PostgreSQL zu konvertieren. Das Erweiterungspaket verwendetAWS LambdaFunktionen zur Emulierung des Verhaltens des SQL Server-Agents.

Sie können ein neues erstellenAWS LambdaFunktion oder registriere eine bestehende Funktion.

So erstellen Sie eine neue AWS Lambda-Funktion

1. InAWS SCT, öffnen Sie im Zieldatenbank-Baum das Kontextmenü (Rechtsklick), wählen SieAnwenden Sie das Erweiterungspaket fürund dannPostgreSQL.

Der Assistent des Erweiterungspakets wird angezeigt.

2. Auf derSQL Server-Agent-Emulationsdienst.Zeigen Sie die Registerkarte an, und gehen Sie wie folgt vor:
 - WähleErstelle einAWS Lambdawirken.
 - FürDatenbank-Login, geben Sie die -Prozedur ein.
 - FürDatenbank-Passwort, geben Sie das Passwort für den Benutzernamen ein, den Sie im vorherigen Schritt eingegeben haben.
 - FürOrdner der Python-Bibliothek, geben Sie den Pfad zu Ihrem Python-Bibliotheksordner ein.
 - Wählen SieErstellenAWS Lambdawirken, und wählen Sie dannAls Nächstes.

Um einAWS LambdaFunktion, die Sie zuvor bereitgestellt haben

- Führen Sie die -Prozedur in der -Datenbank aus.

```
SELECT
  FROM aws_sqlserver_ext.set_service_setting(
    p_service := 'JOB',
    p_setting := 'LAMBDA_ARN',
    p_value := ARN)
```

Im vorherigen Beispiel`ARN`ist die Amazon Resource Name (ARN) der -Prozedur.AWS LambdaFunktion.

Das folgende Beispiel erstellt eine einfache Aufgabe, die aus einem Schritt besteht. Diese Aufgabe führt alle fünf Minuten die zuvor erstellte Aufgabe `awsjob_exampleFunktion`. Diese Funktion fügt Datensätze in die `awsjob_example_table`-Tabelle.

Um diese einfache Aufgabe zu erstellen

1. Erstellen Sie einen Job aus `awsjob_example.sp_add_jobfunktionieren` wie folgt.

```
SELECT
  FROM awsjob_example.sp_add_job (
    par_job_name := 'test_simple_job');
```

2. Erstellen Sie einen Job-Schritt aus `awsjob_example.sp_add_jobstepfunktionieren` wie folgt.

```
SELECT
  FROM awsjob_example.sp_add_jobstep (
    par_job_name := 'test_simple_job',
    par_step_name := 'test_simple_job_step1',
    par_command := 'PERFORM job_simple_example;');
```

Der Job-Schritt spezifiziert, was die Funktion macht.

3. Erstellen Sie einen Scheduler für den Job mit `awsjob_example.sp_add_jobschedulefunktionieren` wie folgt.

```
SELECT
  FROM awsjob_example.sp_add_jobschedule (
    par_job_name := 'test_simple_job',
    par_name := 'test_schedule',
    par_freq_type := 4, /* Daily */
    par_freq_interval := 1, /* frequency_interval is unused */
    par_freq_subday_type := 4, /* Minutes */
    par_freq_subday_interval := 5 /* 5 minutes */);
```

Der Job-Schritt spezifiziert, was die Funktion macht.

Um diesen Job zu löschen, verwenden Sie `awsjob_example.sp_delete_jobfunktionieren` wie folgt.

```
PERFORM awsjob_example.sp_delete_job(
```



```
par_job_name := 'PeriodicJob1'::character varying,  
par_delete_history := 1::smallint,  
par_delete_unused_schedule := 1::smallint);
```

Mit einem AWS SCT Erweiterungspaket zur Emulation von SQL Server-Datenbank-Mail in PostgreSQL

Sie können SQL Server-Datenbank Mail verwenden, um E-Mail-Nachrichten aus der SQL Server-Datenbank-Engine an Benutzer zu senden. Diese E-Mail-Nachrichten können Abfrageergebnisse oder Dateien aus beliebigen Ressourcen in Ihrem Netzwerk enthalten. Weitere Informationen über die SQL Server-Datenbank-E-Mail finden Sie unter [Technische Dokumentation von Microsoft](#).

PostgreSQL hat kein Äquivalent für SQL Server Database Mail. Um die Funktionen von SQL Server Database Mail zu emulieren, AWS SCT erstellt ein Erweiterungspaket. Dieses Erweiterungspaket verwendet AWS Lambda und Amazon Simple Email Service (Amazon SES). AWS Lambda bietet Benutzern eine Schnittstelle zur Interaktion mit dem Amazon SES SES-E-Mail-Versandservice. Um diese Interaktion einzurichten, fügen Sie den Amazon Resource Name (ARN) Ihrer Lambda-Funktion hinzu.

Verwenden Sie für die -Prozedur die -Prozedur.

```
do  
$$  
begin  
PERFORM sysmail_add_account_sp (  
    par_account_name := 'your_account_name',  
    par_email_address := 'your_account_email',  
    par_display_name := 'your_account_display_name',  
    par_mailserver_type := 'AWSLAMBDA'  
    par_mailserver_name := 'ARN'  
);  
end;  
$$ language plpgsql;
```

Verwenden Sie den folgenden Befehl, um den ARN Ihrer Lambda-Funktion zum vorhandenen E-Mail-Konto hinzuzufügen.

```
do  
$$  
begin
```

```
PERFORM sysmail_update_account_sp (  
    par_account_name := 'existind_account_name',  
    par_mailserver_type := 'AWSLAMBDA'  
    par_mailserver_name := 'ARN'  
);  
end;  
$$ language plpgsql;
```

In den vorherigen Beispielen **ARN** ist die ARN Ihrer Lambda-Funktion.

Um das Verhalten von SQL Server Database Mail in PostgreSQL zu emulieren, AWS SCT Die -Prozedur verwendet die folgenden Tabellen, Ansichten und -Prozedur.

Tabellen, die SQL Server-Datenbankmail in PostgreSQL emulieren

Um SQL Server Database Mail zu emulieren, verwendet das Erweiterungspaket die folgenden Tabellen:

sysmail_account

Speichert die -Prozedur.

sysmail_profile

Speichert die -Informationen über die -Prozedur.

sysmail_server

Speichert die -Informationen über die -Prozedur.

sysmail_mailitems

Speichert die Liste der E-Mail-Nachrichten.

sysmail_attachments

Enthält eine -Zeile für die -Prozedur.

sysmail_log

Speichert die Dienstinformationen zum Senden von E-Mail-Nachrichten.

sysmail_profile_account

Speichert die Informationen über die -Prozedur und die -Prozedur.

Ansichten, die SQL Server-Datenbankmail in PostgreSQL emulieren

Um SQL Server-Datenbank-E-Mail zu emulieren, AWS SCT erstellt die folgenden Ansichten in der PostgreSQL-Datenbank, um die Kompatibilität sicherzustellen. Das Erweiterungspaket verwendet sie nicht, aber Ihr konvertierter Code kann diese Ansichten abfragen.

sysmail_allitems

Enthält eine Liste aller E-Mails.

sysmail_faileditems

Enthält eine Liste von E-Mails, die nicht gesendet werden konnten.

sysmail_sentitems

Enthält eine Liste der gesendeten E-Mails.

sysmail_unsentitems

Enthält eine Liste von E-Mails, die noch nicht gesendet wurden.

sysmail_mailattachments

Enthält eine Liste der angehängten Dateien.

Verfahren, die SQL Server-Datenbank-Mail in PostgreSQL emulieren

Um SQL Server Database Mail zu emulieren, verwendet das Erweiterungspaket die folgenden Verfahren:

sp_send_dbmail

Sendet eine E-Mail an die angegebenen Empfänger.

sysmail_add_profile_sp

Erstellt ein neues Benutzerprofil.

sysmail_add_account_sp

Erstellt ein neues E-Mail-Konto, in dem Informationen wie SMTP-Anmeldeinformationen (Simple Mail Transfer Protocol) usw. gespeichert werden.

sysmail_add_profileaccount_sp

Fügt dem angegebenen Benutzerprofil ein E-Mail-Konto hinzu.

sysmail_update_profile_sp

Ändert die Attribute des Benutzerprofils wie Beschreibung, Name usw.

sysmail_update_account_sp

Ändert die Informationen im vorhandenen E-Mail-Konto.

sysmail_update_profileaccount_sp

Aktualisiert die E-Mail-Kontoinformationen im angegebenen Benutzerprofil.

sysmail_delete_profileaccount_sp

Entfernt ein E-Mail-Konto aus dem angegebenen Benutzerprofil.

sysmail_delete_account_sp

Löscht das E-Mail-Konto.

sysmail_delete_profile_sp

Löscht das Benutzerprofil.

sysmail_delete_mailitems_sp

Löscht E-Mails aus internen Tabellen.

sysmail_help_profile_sp

Zeigt Informationen über das Benutzerprofil an.

sysmail_help_account_sp

Zeigt Informationen über das E-Mail-Konto an.

sysmail_help_profileaccount_sp

Zeigt Informationen zu E-Mail-Konten an, die dem Benutzerprofil zugeordnet sind.

sysmail_dbmail_json

Eine interne Prozedur, die JSON-Anfragen generiert für AWS Lambda-Funktionen.

sysmail_verify_profile_sp, sysmail_verify_account_sp, sysmail_verify_addressparams_sp

Interne Verfahren zur Überprüfung der Einstellungen.

sp_get_dbmail, sp_set_dbmail, sysmail_dbmail_xml

Veraltete interne Prozeduren.

Syntax für Prozeduren, die SQL Server-Datenbankmail in PostgreSQL emulieren

Das `aws_sqlserver_ext.sp_send_dbmail` Das Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sp_send_dbmail` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_profile_name varchar = NULL::character varying,
par_recipients text = NULL::text,
par_copy_recipients text = NULL::text,
par_blind_copy_recipients text = NULL::text,
par_subject varchar = NULL::character varying,
par_body text = NULL::text,
par_body_format varchar = NULL::character varying,
par_importance varchar = 'NORMAL'::character varying,
par_sensitivity varchar = 'NORMAL'::character varying,
par_file_attachments text = NULL::text,
par_query text = NULL::text,
par_execute_query_database varchar = NULL::character varying,
par_attach_query_result_as_file smallint = 0,
par_query_attachment_filename varchar = NULL::character varying,
par_query_result_header smallint = 1,
par_query_result_width integer = 256,
par_query_result_separator VARCHAR = ' '::character varying,
par_exclude_query_output smallint = 0,
par_append_query_error smallint = 0,
par_query_no_truncate smallint = 0,
par_query_result_no_padding smallint = 0,
out par_mailitem_id integer,
par_from_address text = NULL::text,
par_reply_to text = NULL::text,
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_delete_mailitems_sp` Das Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_delete_mailitems_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_sent_before timestamp = NULL::timestamp without time zone,
```

```
par_sent_status varchar = NULL::character varying,  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_add_profile_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_add_profile_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_profile_name varchar,  
par_description varchar = NULL::character varying,  
out par_profile_id integer,  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_add_account_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_add_account_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_account_name varchar  
par_email_address varchar  
par_display_name varchar = NULL::character varying  
par_replyto_address varchar = NULL::character varying  
par_description varchar = NULL::character varying  
par_mailserver_name varchar = NULL::character varying  
par_mailserver_type varchar = 'SMTP'::bpchar  
par_port integer = 25  
par_username varchar = NULL::character varying  
par_password varchar = NULL::character varying  
par_use_default_credentials smallint = 0  
par_enable_ssl smallint = 0  
out par_account_id integer  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_add_profileaccount_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_add_profileaccount_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
par_sequence_number integer = NULL::integer,
```

```
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_help_profile_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_help_profile_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_update_profile_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_update_profile_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_description varchar = NULL::character varying,  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_delete_profile_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_delete_profile_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_force_delete smallint = 1,  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_help_account_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_help_account_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_update_account_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_update_account_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
par_email_address varchar = NULL::character varying,  
par_display_name varchar = NULL::character varying,  
par_replyto_address varchar = NULL::character varying,  
par_description varchar = NULL::character varying,  
par_mailserver_name varchar = NULL::character varying,  
par_mailserver_type varchar = NULL::character varying,  
par_port integer = NULL::integer,  
par_username varchar = NULL::character varying,  
par_password varchar = NULL::character varying,  
par_use_default_credentials smallint = NULL::smallint,  
par_enable_ssl smallint = NULL::smallint,  
par_timeout integer = NULL::integer,  
par_no_credential_change smallint = NULL::smallint,  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_delete_account_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_delete_account_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_help_profileaccount_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_help_profileaccount_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_update_profileaccount_sp` Verfahren im Erweiterungspaket emuliert `dasmsdb.dbo.sysmail_update_profileaccount_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).


```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
par_sequence_number integer = NULL::integer,  
out returncode integer
```

Das `aws_sqlserver_ext.sysmail_delete_profileaccount_sp` Verfahren im Erweiterungspaket emuliert das `msdb.dbo.sysmail_delete_profileaccount_sp` Verfahren. Weitere Hinweise zur SQL Server-Datenbank-E-Mail-Prozedur finden Sie unter [Technische Dokumentation von Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

Beispiele für die Verwendung von Prozeduren, die SQL Server-Datenbankmail in PostgreSQL emulieren

Um eine E-Mail zu senden, verwenden Sie die `aws_sqlserver_ext.sp_send_dbmail` Verfahren wie im Folgenden gezeigt.

```
PERFORM sp_send_dbmail (  
    par_profile_name := 'Administrator',  
    par_recipients := 'hello@rusgl.info',  
    par_subject := 'Automated Success Message',  
    par_body := 'The stored procedure finished'  
);
```

Die `sp_send_dbmail` Prozedur zeigt die `sp_send_dbmail` Prozedur mit `par_query`.

```
PERFORM sp_send_dbmail (  
    par_profile_name := 'Administrator',  
    par_recipients := 'hello@rusgl.info',  
    par_subject := 'Account with id = 1',  
    par_query := 'SELECT COUNT(*)FROM Account WHERE id = 1'  
);
```

Die -Prozedur zeigt, wie Sie eine -E-Mail mit HTML-Code senden.

```
DECLARE var_tableHTML TEXT;
SET var_tableHTML := CONCAT(
    '<H1>Work Order Report</H1>',
    '<table border="1">',
    '<tr><th>Work Order ID</th><th>Product ID</th>',
    '<th>Name</th><th>Order Qty</th><th>Due Date</th>',
    '<th>Expected Revenue</th></tr>',
    '</table>'
);
PERFORM sp_send_dbmail (
    par_recipients := 'hello@rusgl.info',
    par_subject := 'Work Order List',
    par_body := var_tableHTML,
    par_body_format := 'HTML'
);
```

Um E-Mails zu löschen, verwenden

Sie `aws_sqlserver_ext.sysmail_delete_mailitems_sp` Verfahren wie im Folgenden gezeigt.

```
DECLARE var_GETDATE datetime;
SET var_GETDATE = NOW();
PERFORM sysmail_delete_mailitems_sp (
    par_sent_before := var_GETDATE
);
```

Die -Prozedur zeigt, wie die ältesten E-Mails gelöscht werden.

```
PERFORM sysmail_delete_mailitems_sp (
    par_sent_before := '31.12.2015'
);
```

Das folgende Beispiel zeigt, wie Sie alle E-Mails löschen können, die nicht gesendet werden können.

```
PERFORM sysmail_delete_mailitems_sp (
    par_sent_status := 'failed'
);
```

Um ein -Profil zu erstellen, verwenden Sie die -

Prozedur `aws_sqlserver_ext.sysmail_add_profile_sp` Verfahren wie im Folgenden gezeigt.

```
PERFORM sysmail_add_profile_sp (  
    profile_name := 'Administrator',  
    par_description := 'administrative mail'  
);
```

Das folgende Beispiel zeigt, wie Sie ein neues Profil erstellen und die eindeutige Profil-ID in einer Variablen speichern.

```
DECLARE var_profileId INT;  
SELECT par_profile_id  
    FROM sysmail_add_profile_sp (  
        profile_name := 'Administrator',  
        par_description := ' Profile used for administrative mail.')
```



```
INTO var_profileId;  
  
SELECT var_profileId;
```

Um eine -Prozedur zu erstellen, verwenden Sie die -
Prozedur `aws_sqlserver_ext.sysmail_add_account_sp` Verfahren wie im Folgenden gezeigt.

```
PERFORM sysmail_add_account_sp (  
    par_account_name := 'Audit Account',  
    par_email_address := 'dba@rusgl.info',  
    par_display_name := 'Test Automated Mailer',  
    par_description := 'Account for administrative e-mail.',  
    par_mailserver_type := 'AWSLAMBDA'  
    par_mailserver_name := 'arn:aws:lambda:us-west-2:555555555555:function:pg_v3'  
);
```

Um dem Benutzerprofil ein E-Mail-Konto hinzuzufügen, verwenden
Sie `aws_sqlserver_ext.sysmail_add_profileaccount_sp` Verfahren wie im Folgenden
gezeigt.

```
PERFORM sysmail_add_profileaccount_sp (  
    par_account_name := 'Administrator',  
    par_account_name := 'Audit Account',  
    par_sequence_number := 1  
);
```

Anwendungsbeispiele für die Emulation von SQL Server-Datenbankmail in PostgreSQL

Wenn Ihr Quelldatenbankcode SQL Server Database Mail zum Senden von E-Mails verwendet, können Sie den AWS SCT Erweiterungspaket, um diesen Code nach PostgreSQL zu konvertieren.

Um eine E-Mail aus Ihrer PostgreSQL-Datenbank zu senden

1. Erstellen und konfigurieren Sie Ihre AWS Lambda Funktion.
2. Wenden Sie den AWS SCT Erweiterungspaket.
3. Erstellen Sie ein Profil über die Prozedur `sysmail_add_profile_sp` funktionieren wie folgt.
4. Erstelle ein E-Mail-Konto über die Prozedur `sysmail_add_account_sp` funktionieren wie folgt.
5. Fügen Sie dieses E-Mail-Konto zu Ihrem Benutzerprofil hinzu, indem `Sysmail_add_profileaccount_sp` funktionieren wie folgt.

```
CREATE OR REPLACE FUNCTION aws_sqlserver_ext.  
proc_dbmail_settings_msdb()  
RETURNS void  
AS  
$BODY$  
BEGIN  
PERFORM aws_sqlserver_ext.sysmail_add_profile_sp(  
    par_profile_name := 'Administrator',  
    par_description := 'administrative mail'  
);  
PERFORM aws_sqlserver_ext.sysmail_add_account_sp(  
    par_account_name := 'Audit Account',  
    par_description := 'Account for administrative e-mail.',  
    par_email_address := 'dba@rusgl.info',  
    par_display_name := 'Test Automated Mailer',  
    par_mailserver_type := 'AWSLAMBDA',  
    par_mailserver_name := 'your_ARN'  
);  
PERFORM aws_sqlserver_ext.sysmail_add_profileaccount_sp(  
    par_profile_name := 'Administrator',  
    par_account_name := 'Audit Account',  
    par_sequence_number := 1  
);  
END;  
$BODY$  
LANGUAGE plpgsql;
```

6. Senden Sie eine E-Mail aus demsp_send_dbmailfunktionieren wie folgt.

```
CREATE OR REPLACE FUNCTION aws_sqlserver_ext.  
proc_dbmail_send_msdb()  
RETURNS void  
AS  
$BODY$  
BEGIN  
PERFORM aws_sqlserver_ext.sp_send_dbmail(  
    par_profile_name := 'Administrator',  
    par_recipients := 'hello@rusgl.info',  
    par_body := 'The stored procedure finished',  
    par_subject := 'Automated Success Message'  
);  
END;  
$BODY$  
LANGUAGE plpgsql;
```

Um die Informationen zu allen Benutzerprofilen einzusehen, verwenden Siesysmail_help_profile_spVerfahren wie im Folgenden gezeigt.

```
SELECT FROM aws_sqlserver_ext.sysmail_help_profile_sp();
```

Das folgende Beispiel zeigt die Informationen zu dem auf der Benutzer an.

```
select from aws_sqlserver_ext.sysmail_help_profile_sp(par_profile_id := 1);  
select from aws_sqlserver_ext.sysmail_help_profile_sp(par_profile_name :=  
'Administrator');
```

Um die Informationen zu allen E-Mail-Konten anzuzeigen, verwenden Sie densysmail_help_account_spVerfahren wie im Folgenden gezeigt.

```
select from aws_sqlserver_ext.sysmail_help_account_sp();
```

Das folgende Beispiel zeigt die Informationen zu dem bestimmten E-Mail-Konto an.

```
select from aws_sqlserver_ext.sysmail_help_account_sp(par_account_id := 1);  
select from aws_sqlserver_ext.sysmail_help_account_sp(par_account_name := 'Audit  
Account');
```

Um die Informationen zu allen E-Mail-Konten anzuzeigen, die mit den Benutzerprofilen verknüpft sind, verwenden Sie `densysmail_help_profileaccount_sp`Verfahren wie im Folgenden gezeigt.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp();
```

Im folgenden Beispiel werden die Datensätze nach ID, Profilname oder Kontoname gefiltert.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_profile_id := 1);
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_profile_id := 1,
  par_account_id := 1);
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_profile_name :=
  'Administrator');
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_account_name := 'Audit
  Account');
```

Um den Namen oder die Beschreibung des Benutzerprofils zu ändern, verwenden Sie `densysmail_update_profile_sp`Verfahren wie im Folgenden gezeigt.

```
select aws_sqlserver_ext.sysmail_update_profile_sp(
  par_profile_id := 2,
  par_profile_name := 'New profile name'
);
```

Um die E-Mail-Kontoeinstellungen zu ändern, verwenden Sie `densysmail_update_account_sp`Verfahren wie im Folgenden gezeigt.

```
select from aws_sqlserver_ext.sysmail_update_account_sp (
  par_account_name := 'Audit Account',
  par_mailserver_name := 'arn:aws:lambda:region:XXXXXXXXXXXX:function:func_test',
  par_mailserver_type := 'AWSLAMBDA'
);
```

Umwandeln von SQL Server in Amazon RDS für SQL Server

Bei der Migration von SQL Server-Schema und -Code in Amazon RDS für SQL Server sind einige Punkte zu beachten:

- AWS SCT kann den SQL Server-Agent konvertieren, um Zeitpläne, Benachrichtigungen und Jobs auf einer Amazon RDS for SQL Server-DB-Instance bereitzustellen. Nach der Umwandlung können

Sie eine Amazon RDS für SQL Server-DB-Instance mit SQL Server Reporting Services (SSRS), SQL Server Analysis Services (SSAS) und SQL Server Integration Services (SSIS) verwenden.

- Amazon RDS unterstützt derzeit SQL Server Service Broker nicht, und auch keine zusätzlichen T-SQL-Endpunkte, für die Sie den Befehl CREATE ENDPOINT ausführen müssen.
- Amazon RDS verfügt über begrenzten Support für verknüpfte Server. Bei der Konvertierung von SQL Server-Anwendungscode, der verknüpfte Server verwendet, AWS SCT konvertiert den Anwendungscode. Überprüfen Sie jedoch das Verhalten von Objekten, die Verbindungsserver verwenden, bevor Sie den umgewandelten Code ausführen.
- AlwaysOn wird verwendet.
- Das AWS SCT Der Bewertungsbericht enthält Servermetriken für die Konvertierung. Zu diesen Metriken über Ihre SQL Server-Instance gehören:
 - Datenspiegelung wird verwendet.
 - SQL Server-Protokollversand ist konfiguriert.
 - Failover-Cluster wird verwendet.
 - Datenbank-E-Mail ist konfiguriert.
 - Full Text Search Service wird verwendet. Amazon RDS für SQL Server verfügt über eine begrenzte Volltextsuche und unterstützt keine semantische Suche.
 - Data Quality Service (DQS) ist installiert. Amazon RDS bietet keine Unterstützung für DQS. Wir raten daher, SQL Server auf einer Amazon EC2-Instance zu installieren.

Rechte für RDS-Prozedur als -Prozedur.

Um zu RDS für SQL Server zu migrieren, erstellen Sie einen Datenbankbenutzer und gewähren Sie dann die erforderlichen Rechte für jede Datenbank. Sie können die -Prozedur verwenden.

```
CREATE LOGIN user_name WITH PASSWORD 'your_password';

USE db_name
CREATE USER user_name FOR LOGIN user_name
GRANT VIEW DEFINITION TO user_name
GRANT VIEW DATABASE STATE TO user_name
GRANT CREATE SCHEMA TO user_name;
GRANT CREATE TABLE TO user_name;
GRANT CREATE VIEW TO user_name;
GRANT CREATE TYPE TO user_name;
GRANT CREATE DEFAULT TO user_name;
GRANT CREATE FUNCTION TO user_name;
```

```
GRANT CREATE PROCEDURE TO user_name;  
GRANT CREATE ASSEMBLY TO user_name;  
GRANT CREATE AGGREGATE TO user_name;  
GRANT CREATE FULLTEXT CATALOG TO user_name;  
GRANT CREATE SYNONYM TO user_name;  
GRANT CREATE XML SCHEMA COLLECTION TO user_name;
```

Im vorherigen Beispiel ersetzen *user_name* mit dem Namen Ihres Benutzers. Dann ersetzen *db_name* mit der -Prozedur -Prozedur. Schließlich ersetzen *dein_Passwort* mit einem sicheren Passwort.

Data Warehouse-Quellen für AWS Schema Conversion Tool

AWS SCT kann Schemas für die folgenden Quell-Data Warehouses in ein unterstütztes Ziel konvertieren. Informationen zu Berechtigungen, Verbindungen und dazu, was für die Verwendung mit der Zieldatenbank oder dem Data Warehouse konvertiert werden AWS SCT kann, finden Sie in den folgenden Details.

Themen

- [Verwendung von Amazon Redshift als Quelle für AWS SCT](#)
- [Verwendung von Azure Synapse Analytics als Quelle für AWS SCT](#)
- [Verwenden von BigQuery als Quelle für AWS SCT](#)
- [Verwendung der Greenplum-Datenbank als Quelle für AWS SCT](#)
- [Verwendung von Netezza als Quelle für AWS SCT](#)
- [Verwenden von Oracle Data Warehouse als Quelle für AWS SCT](#)
- [Verwendung von Snowflake als Quelle für AWS SCT](#)
- [Verwendung von Microsoft SQL Server Data Warehouse als Quelle für AWS SCT](#)
- [Verwendung von Teradata als Quelle für AWS SCT](#)
- [Verwendung von Vertica als Quelle für AWS SCT](#)

Verwendung von Amazon Redshift als Quelle für AWS SCT

Sie können es verwenden AWS SCT, um Ihren Amazon Redshift-Cluster zu optimieren. AWS SCT bietet Ihnen Empfehlungen zur Auswahl von Verteilungs- und Sortierschlüsseln für Ihren Amazon Redshift-Cluster. Sie können das Amazon Redshift-Optimierungsprojekt als ein AWS SCT Projekt betrachten, bei dem Quelle und Ziel auf die verschiedenen Amazon Redshift-Cluster verweisen.

Berechtigungen für Amazon Redshift als Quelldatenbank

Die folgenden Rechte sind erforderlich, um Amazon Redshift als Quelle verwenden zu können:

- USAGE ON SCHEMA *<schema_name>*
- SELECT ON ALL TABLES IN SCHEMA *<schema_name>*
- SELECT ON PG_CATALOG.PG_STATISTIC
- SELECT ON SVV_TABLE_INFO
- SELECT ON TABLE STV_BLOCKLIST
- SELECT ON TABLE STV_TBL_PERM
- WÄHLE SYS_SERVERLESS_USAGE
- WÄHLE AUF PG_DATABASE_INFO
- WÄHLE AUF PG_STATISTIC

Ersetzen Sie in den vorherigen Beispielen den *<schema_name>* Platzhalter durch den Namen des Quellschemas.

Informationen zu den Berechtigungen, die für Amazon Redshift als Ziel erforderlich sind, finden Sie unter [Berechtigungen für Amazon Redshift als Ziel](#).

Verbindung zu Amazon Redshift als Quelle herstellen

Gehen Sie wie folgt vor, um mit dem eine Verbindung zu Ihrer Amazon Redshift-Quelldatenbank herzustellen. AWS Schema Conversion Tool

So stellen Sie eine Verbindung zu einer Amazon Redshift-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Amazon Redshift und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankanmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:

- Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen für die Amazon Redshift-Quelldatenbank einzugeben:

Parameter	Action
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Amazon-Redshift-Datenbank ein.
User name und Password	<p>Geben Sie die Datenbankmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>

Parameter	Action
Use SSL	<p>Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden . Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an:</p> <ul style="list-style-type: none">• Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Trust Stores zu überprüfen.• Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält. Damit dieser Standort hier angezeigt wird, stellen Sie sicher, dass Sie ihn in den globalen Einstellungen hinzufügen. <p>Weitere Informationen zur SSL-Unterstützung für Amazon Redshift finden Sie unter Konfigurieren von Sicherheitsoptionen für Verbindungen.</p>
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.</p>
Redshift-Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projekteinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Amazon Redshift-Optimierungseinstellungen

Um die Amazon Redshift-Optimierungseinstellungen zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Amazon Redshift und dann Amazon Redshift — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Amazon Redshift-Optimierung an.

Die Amazon Redshift-Optimierungseinstellungen AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift-Zielcluster gelten AWS SCT können.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift-Cluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendung von Tabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt es je nach Knotentyp die Anzahl der Tabellen, die auf Ihren Amazon Redshift-Ziel-Cluster angewendet werden sollen. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter [Kontingente und Limits in Amazon Redshift](#) im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert alle Ihre Quelltabellen, auch wenn die Anzahl der Tabellen größer ist, als Ihr Amazon Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie bei der Anwendung des konvertierten Codes das Amazon Redshift-Cluster-Kontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT wendet Tabellen außerdem auf Ihren Amazon Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um die Migrationsstrategie auszuwählen.

AWS empfiehlt, verschiedene Cluster als Quelle und Ziel für Ihr Optimierungsprojekt zu verwenden. Vor dem Start des Amazon Redshift-Optimierungsprozesses erstellen Sie eine Kopie Ihres Amazon Redshift-Quell-Clusters. Sie können Ihre Quelldaten in diese Kopie aufnehmen oder einen leeren Cluster erstellen.

Wählen Sie als Migrationsstrategie Migration to a copy aus, um Daten aus Ihrem Quellcluster in den Zielcluster aufzunehmen.

Wählen Sie unter Migrationsstrategie die Option Migration zu einem Neuanfang aus, um die Optimierungsvorschläge zu überprüfen. Nachdem Sie diese Vorschläge akzeptiert haben, migrieren Sie Ihre Quelldaten zum Zielcluster.

- Um die Komprimierung auf Amazon Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu die Option Kompressionskodierung verwenden aus.

AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift-Algorithmus automatisch eine Kompressionskodierung zu. Weitere Informationen finden Sie unter [Kompressionskodierungen](#) im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Kompressionskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Kompressionskodierung verwenden ausgewählt haben.

- Um mit der automatischen Tabellenoptimierung zu arbeiten.

Die automatische Tabellenoptimierung ist ein Selbstoptimierungsprozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter [Arbeiten mit automatischer Tabellenoptimierung](#) im Amazon Redshift Database Developer Guide.

Wenn Sie es nur für die automatische Tabellenoptimierung verwenden möchten, wählen Sie im linken Bereich Optimierungsstrategien aus. Wählen Sie dann die automatische Tabellenoptimierung von Amazon Redshift verwenden und wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl aus.

- Um die Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

Sie können mithilfe von Amazon Redshift-Metadaten, statistischen Informationen oder diesen beiden Optionen Sortier- und Verteilungsschlüssel auswählen. Wählen Sie auf der Registerkarte Optimierungsstrategien für die erste Schlüsselauswahlstrategie eine der folgenden Optionen aus:

- Metadaten verwenden, statistische Informationen ignorieren
- Metadaten ignorieren, statistische Informationen verwenden
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. AWS SCT definiert anhand dieser Gewichtungswerte, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie Kleine Tabellen finden definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

Sie können nicht nur das Gewicht für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung.

- Geben Sie im Feld Sortierschlüsselspaltenlimit die maximale Anzahl von Spalten in den Sortierschlüssel ein.
- Geben Sie für schiefen Schwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten mit einem Schrägwert über dem Schwellenwert aus der Liste der Kandidaten für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des häufigsten Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top-N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, dessen Abfragestatistiken Sie analysieren möchten.

Verwendung von Azure Synapse Analytics als Quelle für AWS SCT

Sie können AWS SCT verwenden, um Schemas, Codeobjekte und Anwendungscode von Azure Synapse Analytics nach Amazon Redshift zu konvertieren.

Berechtigungen für Azure Synapse Analytics als Quelldatenbank

Die folgenden Berechtigungen sind erforderlich, um ein Azure Synapse Analytics-Data Warehouse als Quelle verwenden zu können:

- VIEW DEFINITION
- VIEW DATABASE STATE

Wenden Sie die Rechte für jede Datenbank an, deren Schema Sie konvertieren.

Verbindung zu Azure Synapse Analytics als Quelle herstellen

Verwenden Sie das folgende Verfahren, um mit dem eine Verbindung zu Ihrem Azure Synapse Analytics-Data Warehouse herzustellen. AWS Schema Conversion Tool

So stellen Sie als Quelle eine Verbindung zu einem Azure Synapse Analytics-Data Warehouse her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Azure Synapse Analytics und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:

- Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:

1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen für das Azure Synapse Analytics Data Warehouse manuell einzugeben:

Parameter	Action
Server name	Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Quelldatenbankservers ein.
SQL-Pool	Geben Sie den Namen des Azure SQL-Pools ein.
User name und Password	<p>Geben Sie die Datenbankanmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>
Use SSL	<p>Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden. Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an:</p> <ul style="list-style-type: none">• Serverzertifikat vertrauen: Wählen Sie diese Option, um dem Serverzertifikat zu vertrauen.• Trust Store: Ein Trust Store, den Sie in den globalen Einstellungen einrichten.
Passwort speichern	AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankkennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einstellungen für die Konvertierung von Azure Synapse Analytics zu Amazon Redshift

Um die Konvertierungseinstellungen von Azure Synapse Analytics in Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Azure Synapse und dann Azure Synapse — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Azure Synapse Analytics zu Amazon Redshift an.

Die Konvertierungseinstellungen von Azure Synapse Analytics zu Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift-Zielcluster gelten AWS SCT können.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift-Cluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendung von Tabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt es je nach Knotentyp die Anzahl der Tabellen, die auf Ihren Amazon Redshift-Ziel-Cluster angewendet werden sollen. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter [Kontingente und Limits in Amazon Redshift](#) im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert alle Ihre Quelltabellen, auch wenn dies mehr ist, als Ihr Amazon Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie bei der Anwendung des konvertierten Codes das

Amazon Redshift-Cluster-Kontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT wendet Tabellen außerdem auf Ihren Amazon Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um Partitionen der Quelltable in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltable erstellt werden AWS SCT können.

Amazon Redshift unterstützt keine Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltable in eine separate Table in Amazon Redshift migrieren. Erstellen Sie AWS SCT dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

AWS SCT bestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltable. Je nach Art der Partitionierung der Quelltable kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon Redshift-Cluster anwenden können. Um zu vermeiden, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltable erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage eines Jahres und zwei Tabellen für NO RANGE und UNKNOWN Partitionen entspricht.

- Um die Komprimierung auf Amazon Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu die Option Kompressionskodierung verwenden aus.

AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift-Algorithmus automatisch eine Kompressionskodierung zu. Weitere Informationen finden Sie unter [Kompressionskodierungen](#) im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Kompressionskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Kompressionskodierung verwenden ausgewählt haben.

Einstellungen für die Konversionsoptimierung von Azure Synapse Analytics zu Amazon Redshift

Um die Einstellungen für die Konversionsoptimierung von Azure Synapse Analytics in Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und anschließend

Konversionseinstellungen. Wählen Sie in der oberen Liste Azure Synapse und dann Azure Synapse — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt die Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Azure Synapse Analytics in Amazon Redshift an.

Die Einstellungen für die Konversionsoptimierung von Azure Synapse Analytics zu Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu die automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

Die automatische Tabellenoptimierung ist ein Selbstoptimierungsprozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter [Arbeiten mit automatischer Tabellenoptimierung](#) im Amazon Redshift Database Developer Guide.

Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl aus.

- Um die Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

Sie können mithilfe von Amazon Redshift-Metadaten, statistischen Informationen oder diesen beiden Optionen Sortier- und Verteilungsschlüssel auswählen. Wählen Sie auf der Registerkarte Optimierungsstrategien für die erste Schlüsselauswahlstrategie eine der folgenden Optionen aus:

- Metadaten verwenden, statistische Informationen ignorieren
- Metadaten ignorieren, statistische Informationen verwenden
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. AWS SCT definiert anhand dieser Gewichtungswerte, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie Kleine Tabellen finden definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

Sie können nicht nur das Gewicht für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung.

- Geben Sie im Feld Sortierschlüsselspaltenlimit die maximale Anzahl von Spalten in den Sortierschlüssel ein.
- Geben Sie für schiefen Schwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCTschließt Spalten, deren Verzerrungswert größer als der Schwellenwert ist, aus der Liste der Kandidaten für den Verteilungsschlüssel aus. AWS SCTdefiniert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des häufigsten Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top-N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Auf der Registerkarte Optimierungsstrategien können Sie auch die Größe kleiner Tabellen für die Strategie Kleine Tabellen suchen definieren. Geben Sie für Minimale Tabellenzeilenanzahl und Maximale Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCTwendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verwenden von BigQuery als Quelle für AWS SCT

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Amazon Redshift BigQuery nach Amazon Redshift zu konvertieren.

Rechte für BigQuery als Quelle

Um ein BigQuery Data Warehouse als Quelle in zu verwendenAWS SCT, erstellen Sie ein Servicekonto. In Google Cloud verwenden Anwendungen Dienstkonten, um autorisierte API-Aufrufe zu tätigen. Dienstkonten unterscheiden sich von Benutzerkonten. Weitere Informationen finden Sie unter [Dienstkonten](#) in der Google Cloud Identity and Access Management-Dokumentation.

Stellen Sie sicher, dass Sie Ihrem Dienstkonto die folgenden Rollen zuweisen:

- BigQuery Admin
- Storage Admin

Die BigQuery Admin Rolle bietet Berechtigungen zur Verwaltung aller Ressourcen innerhalb des Projekts. AWS SCT verwendet diese Rolle, um Ihre BigQuery Metadaten in das Migrationsprojekt zu laden.

Die Storage Admin Rolle gewährt die volle Kontrolle über Datenobjekte und Buckets. Sie finden diese Rolle unter Cloud Storage. AWS SCT verwendet diese Rolle, um Ihre Daten aus Amazon Redshift zu extrahieren BigQuery und sie dann in Amazon Redshift zu laden.

Um eine Schlüsseldatei für ein Dienstkonto zu erstellen

1. Melden Sie sich bei der Google Cloud-Managementkonsole unter <https://console.cloud.google.com/> an.
2. Wählen Sie auf der [BigQueryAPI-Seite](#) die Option Aktivieren aus. Überspringen Sie diesen Schritt, wenn API aktiviert angezeigt wird.
3. Wählen Sie auf der Seite [Dienstkonten](#) Ihr Projekt aus und klicken Sie dann auf Dienstkonto erstellen.
4. Geben Sie auf der Seite mit den Servicekontodetails einen beschreibenden Wert für den Dienstkontonamen ein. Wählen Sie Erstellen und fahren Sie fort. Die Seite Diesem Dienstkonto Zugriff auf das Projekt gewähren wird geöffnet.
5. Wählen Sie für „Rolle auswählen“ BigQuery die Option und dann BigQueryAdmin aus.
6. Wählen Sie Weitere Rolle hinzufügen. Wählen Sie für Select a role die Option Cloud Storage und dann Storage Admin aus.
7. Wählen Sie Weiter und dann Fertig.
8. Wählen Sie auf der Seite [Dienstkonten](#) das Dienstkonto aus, das Sie erstellt haben.
9. Wählen Sie Schlüssel und wählen Sie dann Neuen Schlüssel erstellen für Schlüssel hinzufügen aus.
10. Wählen Sie JSON und dann Erstellen aus. Wählen Sie den Ordner zum Speichern Ihres privaten Schlüssels oder wählen Sie den Standardordner für Downloads in Ihrem Browser.

Verwenden Sie den Google Cloud Storage-Bucket-Ordner, AWS SCT um BigQuery Daten aus einem Data Warehouse zu extrahieren. Erstellen Sie diesen Bucket, bevor Sie mit der Datenmigration beginnen. Geben Sie im Dialogfeld „Lokale Aufgabe erstellen“ den Pfad zu Ihrem Google Cloud Storage-Bucket-Ordner ein. Weitere Informationen finden Sie unter [Eine AWS SCT Aufgabe erstellen, ausführen und überwachen](#).

Verbindung BigQuery als Quelle herstellen

Gehen Sie wie folgt vor, um mit dem eine Verbindung zu Ihrem BigQuery Quellprojekt herzustellenAWS Schema Conversion Tool.

So stellen Sie eine Verbindung zu einem BigQuery Quell-Data Warehouse her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie BigQueryund dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihr BigQuery Projekt ein. AWS SCTzeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Geben Sie unter Schlüsselpfad den Pfad zur Schlüsseldatei des Dienstkontos ein. Weitere Hinweise zum Erstellen dieser Datei finden Sie unter[Rechte für BigQuery als Quelle](#).
5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrem BigQuery Quellprojekt hergestellt werden AWS SCT kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrem BigQuery Quellprojekt herzustellen.

Einschränkungen bei der Verwendung BigQuery als Quelle für AWS SCT

Die folgenden Einschränkungen gelten bei der Verwendung BigQuery als Quelle fürAWS SCT:

- AWS SCTunterstützt nicht die Konvertierung von Unterabfragen in analytische Funktionen.
- Sie können keine `SELECT AS VALUE` Anweisungen AWS SCT zum Umrechnen BigQuery `SELECT AS STRUCT` verwenden.
- AWS SCTunterstützt die Konvertierung der folgenden Funktionstypen nicht:
 - Ungefähre Summe
 - Bit
 - Debugging
 - Verbundabfrage
 - Geografie
 - Hash
 - Mathematisch
 - Netto

- Statistisches Aggregat
- UUID
- AWS SCT bietet eingeschränkte Unterstützung für die Konvertierung von Zeichenkettenfunktionen.
- AWS SCT unterstützt die Konvertierung von UNNEST Operatoren nicht.
- Sie können korrelierte Join-Operationen nicht in AWS SCT konvertieren.
- AWS SCT unterstützt die Konvertierung von QUALIFY, WINDOWLIMIT, und OFFSET -Klauseln nicht.
- Sie können rekursive allgemeine Tabellenausdrücke nicht verwenden AWS SCT.
- AWS SCT unterstützt nicht die Konvertierung von INSERT Anweisungen mit Unterabfragen innerhalb von VALUES Klauseln.
- AWS SCT unterstützt die Konvertierung von UPDATE Anweisungen für verschachtelte Felder und wiederholte Datensätze nicht.
- Sie können ARRAY Datentypen nicht AWS SCT STRUCT zum Konvertieren verwenden.

BigQuery zu Amazon Redshift-Konvertierungseinstellungen

Um die Konvertierungseinstellungen von Amazon Redshift BigQuery zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Google BigQuery und dann Google BigQuery — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die BigQuery Konvertierung in Amazon Redshift an.

BigQuery Die Konvertierungseinstellungen zu Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift-Zielcluster gelten AWS SCT können.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift-Cluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendung von Tabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt es je nach Knotentyp die Anzahl der Tabellen, die auf Ihren Amazon Redshift-Ziel-Cluster angewendet werden sollen. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter [Kontingente und Limits in Amazon Redshift](#) im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert alle Ihre Quelltabellen, auch wenn dies mehr ist, als Ihr Amazon Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie bei der Anwendung des konvertierten Codes das Amazon Redshift-Cluster-Kontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT wendet Tabellen außerdem auf Ihren Amazon Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um die Komprimierung auf Amazon Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu die Option Kompressionskodierung verwenden aus.

AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift-Algorithmus automatisch eine Kompressionskodierung zu. Weitere Informationen finden Sie unter [Kompressionskodierungen](#) im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Kompressionskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Kompressionskodierung verwenden ausgewählt haben.

BigQuery zu den Einstellungen für die Konvertierungsoptimierung von Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von Amazon Redshift BigQuery zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Google BigQuery und dann Google BigQuery — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt die Einstellungen zur Konvertierungsoptimierung für BigQuery die Konvertierung in Amazon Redshift an.

BigQuery zu den Einstellungen für die Konvertierungsoptimierung zu Amazon Redshift AWS SCT gehören Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu die automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

Die automatische Tabellenoptimierung ist ein Selbstoptimierungsprozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter [Arbeiten mit automatischer Tabellenoptimierung](#) im Amazon Redshift Database Developer Guide.

Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl aus.

- Um die Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

Sie können mithilfe von Amazon Redshift-Metadaten, statistischen Informationen oder diesen beiden Optionen Sortier- und Verteilungsschlüssel auswählen. Wählen Sie auf der Registerkarte Optimierungsstrategien für die erste Schlüsselauswahlstrategie eine der folgenden Optionen aus:

- Metadaten verwenden, statistische Informationen ignorieren
- Metadaten ignorieren, statistische Informationen verwenden
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0–100) ein. Diese Werte definieren das Gewicht jeder Strategie. AWS SCT definiert anhand dieser Gewichtungswerte, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie Kleine Tabellen finden definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

Sie können nicht nur das Gewicht für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung.

- Geben Sie im Feld Sortierschlüsselspaltenlimit die maximale Anzahl von Spalten in den Sortierschlüssel ein.

- Geben Sie für schiefen Schwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCTschließt Spalten, deren Verzerrungswert größer als der Schwellenwert ist, aus der Liste der Kandidaten für den Verteilungsschlüssel aus. AWS SCTdefiniert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des häufigsten Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top-N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Auf der Registerkarte Optimierungsstrategien können Sie auch die Größe kleiner Tabellen für die Strategie Kleine Tabellen suchen definieren. Geben Sie für Minimale Tabellenzeilenanzahl und Maximale Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCTwendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verwendung der Greenplum-Datenbank als Quelle für AWS SCT

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Greenplum Database nach Amazon Redshift zu konvertieren.

Rechte für Greenplum Database als Quelle

Die folgenden Rechte sind erforderlich, um Greenplum Database als Quelle verwenden zu können:

- CONNECT ON DATABASE *<database_name>*
- USAGE ON SCHEMA *<schema_name>*
- WÄHLEN SIE EIN AUS *<schema_name>*. *<table_name>*
- WÄHLEN SIE EINE SEQUENZ AUS *<schema_name>*. *<sequence_name>*

Ersetzen Sie die Platzhalter im vorherigen Beispiel wie folgt:

- Ersetzen Sie *database_name* durch den Namen der Quelldatenbank.
- Ersetzen Sie *schema_name* durch den Namen des Quellschemas.
- Ersetzen Sie *table_name* durch den Namen der Quelltable.
- Ersetzen Sie *sequence_name* durch den Namen des Sequenznamens.

Verbindung zur Greenplum-Datenbank als Quelle herstellen

Gehen Sie wie folgt vor, um eine Verbindung zu Ihrer Greenplum-Quelldatenbank herzustellen AWS SCT.

So stellen Sie eine Verbindung zu einer Greenplum-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie SAP ASE und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankanmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um die Datenbankanmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankanmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Anmeldeinformationen der Greenplum-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Greenplum-Datenbank ein.

Parameter	Action
User name und Password	<p>Geben Sie die Datenbankanmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>
Use SSL	<p>Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden . Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an:</p> <ul style="list-style-type: none">• Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Trust Stores zu überprüfen.• Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält.
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankkennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.</p>

Parameter	Action
Treiberpfad für die Greenplum-Datenbank	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projekteinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einstellungen für die Konvertierung von Greenplum nach Amazon Redshift

Um die Konvertierungseinstellungen von Greenplum nach Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Greenplum und dann Greenplum — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Greenplum in Amazon Redshift an.

Die Konvertierungseinstellungen von Greenplum nach Amazon Redshift in AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihrem konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift-Zielcluster gelten, wählen Sie die maximale Anzahl von Tabellen aus.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift-Cluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendung von Tabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt es je nach Knotentyp die Anzahl der Tabellen, die auf Ihren Amazon Redshift-Ziel-Cluster angewendet werden sollen. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter [Kontingente und Limits in Amazon Redshift](#) im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert alle Ihre Quelltabellen, auch wenn dies mehr ist, als Ihr Amazon Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie bei der Anwendung des konvertierten Codes das Amazon Redshift-Cluster-Kontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT wendet Tabellen außerdem auf Ihren Amazon Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um Partitionen der Quelltable in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltable erstellt werden AWS SCT können.

Amazon Redshift unterstützt keine Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltable in eine separate Tabelle in Amazon Redshift migrieren. Erstellen Sie AWS SCT dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

AWS SCT bestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltable. Je nach Art der Partitionierung der Quelltable kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon Redshift-Cluster anwenden können. Um zu vermeiden, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltable erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage eines Jahres und zwei Tabellen für NO RANGE und UNKNOWN Partitionen entspricht.

- Um die Komprimierung auf Amazon Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu die Option Kompressionskodierung verwenden aus.

AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift-Algorithmus automatisch eine Kompressionskodierung zu. Weitere Informationen finden Sie unter [Kompressionskodierungen](#) im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Kompressionskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Kompressionskodierung verwenden ausgewählt haben.

Einstellungen für die Konvertierungsoptimierung von Greenplum nach Amazon Redshift

Um die Einstellungen für die Konversionsoptimierung von Greenplum nach Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Greenplum und dann Greenplum — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt die Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Greenplum nach Amazon Redshift an.

Die Einstellungen zur Konvertierungsoptimierung von Greenplum nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu die automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

Die automatische Tabellenoptimierung ist ein Selbstoptimierungsprozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter [Arbeiten mit automatischer Tabellenoptimierung](#) im Amazon Redshift Database Developer Guide.

Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl aus.

- Um die Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

Sie können mithilfe von Amazon Redshift-Metadaten, statistischen Informationen oder diesen beiden Optionen Sortier- und Verteilungsschlüssel auswählen. Wählen Sie auf der Registerkarte Optimierungsstrategien für die erste Schlüsselauswahlstrategie eine der folgenden Optionen aus:

- Metadaten verwenden, statistische Informationen ignorieren
- Metadaten ignorieren, statistische Informationen verwenden
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. AWS SCT definiert anhand dieser Gewichtungswerte, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie Kleine Tabellen finden definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

Sie können nicht nur das Gewicht für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung.

- Geben Sie im Feld Sortierschlüsselspaltenlimit die maximale Anzahl von Spalten in den Sortierschlüssel ein.
- Geben Sie für schiefen Schwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, deren Verzerrungswert größer als der Schwellenwert ist, aus der Liste der Kandidaten für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des häufigsten Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top-N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Auf der Registerkarte Optimierungsstrategien können Sie auch die Größe kleiner Tabellen für die Strategie Kleine Tabellen suchen definieren. Geben Sie für Minimale Tabellenzeilenanzahl und Maximale Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verwendung von Netezza als Quelle für AWS SCT

Sie können AWS SCT verwenden, um Schemas, Codeobjekte und Anwendungscode von Netezza nach Amazon Redshift zu konvertieren.

Privilegien für Netezza als Quelle

Die folgenden Rechte sind erforderlich, um Netezza als Quelle verwenden zu können:

- wählen Sie in der `system.definition_schema.system`-Ansicht
- wählen Sie in der Tabelle `system.definition_schema.system` aus
- wählen Sie in der Tabelle `system.definition_schema.management`
- liste auf `<database_name>`
- liste auf `<schema_name>`
- Liste auf `<database_name>.all.table`
- Liste in der Tabelle `<database_name>.all.external`
- Liste auf `<database_name>.all.view`
- Liste in der Ansicht „`<database_name>.all.materialized`“
- Liste auf `<database_name>.all.procedure`
- Liste auf `<database_name>.all.sequence`
- Liste auf `<database_name>.all.function`
- Liste auf `<database_name>.all.aggregate`

Ersetzen Sie die Platzhalter im vorherigen Beispiel wie folgt:

- Ersetzen Sie *database_name* durch den Namen der Quelldatenbank.
- Ersetzen Sie *schema_name* durch den Namen des Quellschemas.

AWS SCT erfordert Zugriff auf die folgenden Systemtabellen und Ansichten. Sie können Zugriff auf diese Objekte gewähren, anstatt Zugriff auf `system.definition_schema.system view` und `system.definition_schema.system tables` in der vorherigen Liste zu gewähren.

- wählen Sie `system.definition_schema` aus. `_t_aggregat`
- wählen Sie `system.definition_schema` aus. `_t_class`
- wählen Sie `system.definition_schema` aus. `_t_Beschränkung`

- wählen Sie system.definition_schema aus. _t_const_relattr
- wählen Sie system.definition_schema aus. _t_database
- wählen Sie system.definition_schema aus. _t_grpobj_priv
- wählen Sie system.definition_schema aus. _t_grpusr
- wählen Sie system.definition_schema aus. _t_hist_config
- wählen Sie system.definition_schema aus. _t_objekt
- wählen Sie system.definition_schema aus. _t_objekt_klassen
- wählen Sie system.definition_schema aus. _t_proc
- wählen Sie system.definition_schema aus. _t_typ
- wählen Sie system.definition_schema aus. _t_user
- wählen Sie system.definition_schema aus. _t_usobj_priv
- wählen Sie system.definition_schema aus. _vt_sequenz
- wählen Sie system.definition_schema aus. _v_aggregat
- wählen Sie system.definition_schema aus. _v_constraint_depends
- wählen Sie system.definition_schema aus. _v_database
- wählen Sie system.definition_schema aus. _v_datatype
- wählen Sie system.definition_schema aus. _v_dslice
- wählen Sie system.definition_schema aus. _v_function
- wählen Sie system.definition_schema aus. _v_gruppe
- wählen Sie system.definition_schema aus. _v_obj_relation
- wählen Sie system.definition_schema aus. _v_obj_relation_xdb
- wählen Sie system.definition_schema aus. _v_procedure
- wählen Sie system.definition_schema aus. _v_relation_column
- wählen Sie system.definition_schema aus. _v_relation_keydata
- wählen Sie system.definition_schema aus. _v_relobjclasses
- wählen Sie system.definition_schema aus. _v_schema_xdb
- wählen Sie system.definition_schema aus. _v_sequenz
- wählen Sie system.definition_schema aus. _v_synonym
- wählen Sie system.definition_schema aus. _v_system_info
- wählen Sie system.definition_schema aus. _v_sys_constraint

- wählen Sie system.definition_schema aus. _v_sys_object_dslice_info
- wählen Sie system.definition_schema aus. _v_sys_user
- wählen Sie system.definition_schema aus. _v_table
- wählen Sie system.definition_schema aus. _v_table_constraint
- wählen Sie system.definition_schema aus. _v_table_dist_map
- wählen Sie system.definition_schema aus. _v_table_organize_column
- wählen Sie system.definition_schema aus. _v_table_storage_stat
- wählen Sie system.definition_schema aus. _v_user
- wählen Sie system.definition_schema aus. _v_view
- wählen Sie system.information_schema aus. _v_relation_column
- wählen Sie system.information_schema aus. _v_table
- wähle auf \$hist_column_access_*

Verbindung zu Netezza als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Netezza-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Netezza-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Netezza und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen der Netezza-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
User name und Password	<p>Geben Sie die Datenbankmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>
Passwort speichern	AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankkennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.

Parameter	Action
Netezza-Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projekteinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Konfiguration der laufenden Datenreplikation

Nachdem Sie Ihre Netezza-Datenbankschemas konvertiert und auf Ihre Amazon Redshift-Datenbank angewendet haben, können Sie Daten mit AWS SCT Datenextraktionsagenten migrieren. Der Agent extrahiert Ihre Daten und lädt sie in Ihren Amazon S3-Bucket hoch. Sie können die Daten dann verwenden, um die Daten von Amazon S3 nach Amazon Redshift zu kopieren.

Wenn sich die Daten in Ihrer Quelldatenbank während des Migrationsprozesses ändern, können Sie laufende Änderungen mit Ihren AWS SCT Datenextraktionsagenten erfassen. Anschließend können Sie diese laufenden Änderungen in Ihrer Zieldatenbank replizieren, nachdem Sie die erste Datenmigration abgeschlossen haben. Dieser Vorgang wird als fortlaufende Datenreplikation oder Change Data Capture (CDC) bezeichnet.

Um die laufende Datenreplikation für Migrationen von Netezza zu Amazon Redshift zu konfigurieren

1. Erstellen Sie in Ihrer Quelldatenbank eine Verlaufsdatenbank. Sie können das folgende Codebeispiel in der Netezza-Befehlszeilenschnittstelle (CLI) verwenden.

```
nzhistcreatedb -d history_database_name -t query -v 1 -u load_user -o histdb_owner
-p your_password
```

Ersetzen Sie im vorherigen Beispiel *history_database_name* durch den Namen Ihrer Verlaufsdatenbank. Ersetzen Sie anschließend *load_user* durch den Namen des Benutzers, den Sie definiert haben, um Verlaufsdaten in die Datenbank zu laden. Ersetzen Sie dann *histdb_owner* durch den Namen des Benutzers, den Sie als Besitzer der Verlaufsdatenbank definiert haben. Stellen Sie sicher, dass Sie diesen Benutzer bereits erstellt und die CREATE DATABASE Berechtigung erteilt haben. Ersetzen Sie *abschließend your_password* durch ein sicheres Passwort.

2. Konfigurieren Sie die Verlaufsprotokollierung. Verwenden Sie dazu das folgende Codebeispiel.

```
CREATE HISTORY CONFIGURATION history_configuration_name HISTTYPE QUERY
  DATABASE history_database_name USER load_user PASSWORD your_password COLLECT
  PLAN, COLUMN
  LOADINTERVAL 1 LOADMINTHRESHOLD 0 LOADMAXTHRESHOLD 0 STORAGELIMIT 25
  LOADRETRY 2 VERSION 1;
```

Ersetzen Sie im vorherigen Beispiel *history_configuration_name* und *history_database_name* durch die Namen Ihrer Verlaufsdatenbank und Ihrer Verlaufsdatenbank. Ersetzen Sie anschließend *load_user* durch den Namen des Benutzers, den Sie definiert haben, um Verlaufsdaten in die Datenbank zu laden. Ersetzen Sie *dann your_password* durch ein sicheres Passwort.

3. Erteilen Sie Leseberechtigungen für alle Tabellen in der Verlaufsdatenbank. Sie können das folgende Codebeispiel verwenden, um die SELECT Erlaubnis zu erteilen.

```
GRANT SELECT ON history_database_name.ALL.TABLE TO your_user;
```

Ersetzen Sie im vorherigen Beispiel *history_database_name* durch den Namen Ihrer Verlaufsdatenbank. Ersetzen Sie *anschließend your_user* durch den Namen des Benutzers mit minimalen Berechtigungen für die Arbeit mit Ihrer Netezza-Datenbank. Sie verwenden die Anmeldeinformationen dieses Datenbankbenutzers in AWS SCT.

4. Erfassen Sie Statistiken für jede Tabelle in Ihrem Quellschema, um Informationen über die Kardinalität von Spalten zu erhalten. Sie können den folgenden Befehl verwenden, um Statistiken in Ihrer Verlaufsdatenbank zu generieren.

```
GENERATE STATISTICS on "schema_name". "table_name";
```

Ersetzen Sie im vorherigen Beispiel *schema_name* und *table_name* durch den Namen Ihres Datenbankschemas und Ihrer Tabelle.

5. Stellen Sie sicher, dass Sie die Voraussetzungen erfüllt haben, indem Sie die folgende Abfrage ausführen:

```
SELECT COUNT(*)
FROM history_database_name.history_schema_name."$hist_column_access_N";
```

Ersetzen Sie im vorherigen Beispiel *history_database_name* und *history_schema_name* durch den Namen Ihrer Verlaufsdatenbank und Ihres Schemas. Als Nächstes ersetzen Sie *N* durch die Versionsnummer Ihrer Verlaufsdatenbank. Weitere Informationen zu historischen Datenbankversionen finden Sie in der [IBM Netezza-Dokumentation](#).

6. Installieren Sie Ihre Datenextraktionsagenten. Weitere Informationen finden Sie unter [Installation von Extraktionsagenten](#).

Stellen Sie sicher, dass der {working.folder} Parameter in der settings.properties Datei für alle Extractor-Instanzen auf denselben Ordner verweist. In diesem Fall können Ihre Extraktoren die CDC-Sitzung koordinieren und einen einzigen Transaktionspunkt für alle Unteraufgaben verwenden.

7. Registrieren Sie Ihren Datenextraktionsagenten. Weitere Informationen finden Sie unter [Registrierung von Extraktionsagenten mit dem AWS Schema Conversion Tool](#).
8. Erstellen Sie Ihre CDC-Aufgabe. Weitere Informationen finden Sie unter [Eine AWS SCT Aufgabe erstellen, ausführen und überwachen](#).
 - a. Öffnen Sie Ihr Projekt in AWS SCT. Wählen Sie im linken Bereich Ihre Quelltable aus. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Lokale Aufgabe erstellen.
 - b. Geben Sie unter Aufgabenname einen aussagekräftigen Namen für Ihre Datenmigrationsaufgabe ein.
 - c. Wählen Sie für den Migrationsmodus Extrahieren, Hochladen und Kopieren aus.
 - d. Wählen Sie CDC aktivieren aus.
 - e. Wählen Sie die Registerkarte CDC-Einstellungen und definieren Sie den Umfang und den Zeitplan der CDC-Sitzungen.
 - f. Wählen Sie Testaufgabe, um zu überprüfen, ob Sie eine Verbindung zu Ihrem Arbeitsordner, Amazon S3-Bucket und Amazon Redshift Data Warehouse herstellen können.

- g. Wählen Sie Erstellen, um Ihre Aufgabe zu erstellen.
 - h. Wählen Sie den Tab Aufgaben, wählen Sie Ihre Aufgabe aus der Liste aus und wählen Sie Start.
9. Die AWS SCT Aufgabe gewährleistet die Transaktionskonsistenz in der Zieldatenbank. Der Datenextraktionsagent repliziert Transaktionen aus der Quelle in der Reihenfolge der Transaktions-ID.

Wenn Sie eine der Migrationssitzungen beenden oder diese fehlschlägt, wird auch die CDC-Verarbeitung gestoppt.

Einstellungen für die Konvertierung von Netezza nach Amazon Redshift

Um die Konvertierungseinstellungen von Netezza nach Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Netezza und dann Netezza — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Netezza nach Amazon Redshift an.

Die Konvertierungseinstellungen von Netezza nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift-Zielcluster gelten AWS SCT können.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift-Cluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendung von Tabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt es je nach Knotentyp die Anzahl der Tabellen, die auf Ihren Amazon Redshift-Ziel-Cluster angewendet

werden sollen. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter [Kontingente und Limits in Amazon Redshift](#) im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert alle Ihre Quelltabellen, auch wenn dies mehr ist, als Ihr Amazon Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie bei der Anwendung des konvertierten Codes das Amazon Redshift-Cluster-Kontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT wendet Tabellen außerdem auf Ihren Amazon Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um die Komprimierung auf Amazon Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu die Option Kompressionskodierung verwenden aus.

AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift-Algorithmus automatisch eine Kompressionskodierung zu. Weitere Informationen finden Sie unter [Kompressionskodierungen](#) im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Kompressionskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Kompressionskodierung verwenden ausgewählt haben.

Einstellungen für die Konversionsoptimierung von Netezza nach Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von Netezza zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Netezza und dann Netezza — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt die Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Netezza nach Amazon Redshift an.

Die Einstellungen zur Konvertierungsoptimierung von Netezza nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu die automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

Die automatische Tabellenoptimierung ist ein Selbstoptimierungsprozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter [Arbeiten mit automatischer Tabellenoptimierung](#) im Amazon Redshift Database Developer Guide.

Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl aus.

- Um die Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

Sie können mithilfe von Amazon Redshift-Metadaten, statistischen Informationen oder diesen beiden Optionen Sortier- und Verteilungsschlüssel auswählen. Wählen Sie auf der Registerkarte Optimierungsstrategien für die erste Schlüsselauswahlstrategie eine der folgenden Optionen aus:

- Metadaten verwenden, statistische Informationen ignorieren
- Metadaten ignorieren, statistische Informationen verwenden
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. AWS SCT definiert anhand dieser Gewichtungswerte, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie Kleine Tabellen finden definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

Sie können nicht nur das Gewicht für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung.

- Geben Sie im Feld Sortierschlüsselspaltenlimit die maximale Anzahl von Spalten in den Sortierschlüssel ein.
- Geben Sie für schiefen Schwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, deren Verzerrungswert größer als der Schwellenwert ist, aus der Liste der Kandidaten für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen

Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des häufigsten Werts zur Gesamtzahl der Datensätze.

- Geben Sie für die Top-N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Auf der Registerkarte Optimierungsstrategien können Sie auch die Größe kleiner Tabellen für die Strategie Kleine Tabellen suchen definieren. Geben Sie für Minimale Tabellenzeilenanzahl und Maximale Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCTwendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verwenden von Oracle Data Warehouse als Quelle für AWS SCT

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Oracle Data Warehouse in Amazon Redshift oder Amazon Redshift zu konvertieren und in Kombination zu AWS Glue verwenden.

Rechte für Oracle Data Warehouse als Quelle

Die folgenden Rechte sind erforderlich, um Oracle Data Warehouse als Quelle verwenden zu können:

- connect
- select_catalog_role
- select any dictionary

Verbindung zu Oracle Data Warehouse als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit den AWS Schema Conversion Tool eine Verbindung zu Ihrer Oracle Data Warehouse-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Oracle Data Warehouse-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Oracle und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Oracle Source Data Warehouse-Verbindungsinformationen manuell einzugeben:

Parameter	Action
Typ	<p>Wählen Sie den Verbindungstyp für die Datenbank aus. Geben Sie abhängig vom gewählten Typ die folgenden zusätzlichen Informationen ein:</p> <ul style="list-style-type: none"> • SID <ul style="list-style-type: none"> • Servername: Der DNS-Name (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers. • Server port: Der Port, der für die Verbindung zum Quelldatenbankserver verwendet wird • Oracle SID: Die Oracle-System-ID (SID). Um die Oracle-SID abzurufen, senden Sie die folgende Abfrage an Ihre Oracle-Datenbank: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre> • Service-Name <ul style="list-style-type: none"> • Server name: Der DNS-Name oder die IP-Adresse Ihres Quelldatenbankservers

Parameter	Action
	<ul style="list-style-type: none">• Server port: Der Port, der für die Verbindung zum Quelldatenbankserver verwendet wird• Service Name: Der Name des Oracle-Services, zu dem eine Verbindung hergestellt werden soll• TNS-Alias<ul style="list-style-type: none">• TNS file path: Der Pfad zu der Datei, die die Transparent Network Substrate (TNS)-Namensverbindungsinformationen enthält• TNS file path: Der TNS-Alias aus der Datei, die für die Verbindung zur Quelldatenbank verwendet wird• TNS Connect-ID<ul style="list-style-type: none">• TNS-Verbindungs-ID: Die Kennung für die registrierten TNS-Verbindungsinformationen.
User name und Password	<p>Geben Sie die Datenbankmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>

Parameter	Action
Use SSL	<p>Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden . Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an:</p> <ul style="list-style-type: none"> • SSL-Authentifizierung: Wählen Sie diese Option, um die SSL-Authentifizierung für die Verbindung zu verwenden. • Trust Store: Der Speicherort eines Trust Stores, der Zertifikate enthält. • Schlüsselspeicher: Der Standort eines Schlüssel speichers, der einen privaten Schlüssel und Zertifikate enthält. Dieser Wert ist erforderlich, wenn die SSL-Authentifizierung ausgewählt ist, und ist ansonsten optional.
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.</p>
Oracle-Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projekteinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einstellungen für die Konvertierung von Oracle Data Warehouse nach Amazon Redshift

Um die Konvertierungseinstellungen von Oracle Data Warehouse zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Oracle und dann Oracle — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Oracle Data Warehouse zu Amazon Redshift an.

Die Konvertierungseinstellungen von Oracle Data Warehouse nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift-Zielcluster gelten AWS SCT können.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift-Cluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendung von Tabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt es je nach Knotentyp die Anzahl der Tabellen, die auf Ihren Amazon Redshift-Ziel-Cluster angewendet werden sollen. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter [Kontingente und Limits in Amazon Redshift](#) im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert alle Ihre Quelltabellen, auch wenn dies mehr ist, als Ihr Amazon Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie bei der Anwendung des konvertierten Codes das Amazon Redshift-Cluster-Kontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT wendet Tabellen außerdem auf Ihren Amazon Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um Partitionen der Quelltable in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltable erstellt werden AWS SCT können.

Amazon Redshift unterstützt keine Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltable in eine separate Table in Amazon Redshift migrieren. Erstellen Sie AWS SCT dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

AWS SCTbestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltable. Je nach Art der Partitionierung der Quelltable kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon Redshift-Cluster anwenden können. Um zu vermeiden, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltable erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage eines Jahres und zwei Tabellen für NO RANGE und UNKNOWN Partitionen entspricht.

- Um die Daten zu konvertieren, geben Sie Formatierungsfunktionen wie TO_CHAR TO_DATE, und TO_NUMBER mit Datetime-Formatelementen ein, die Amazon Redshift nicht unterstützt. AWS SCTVerwendet standardmäßig die Funktionen des Erweiterungspakets, um die Verwendung dieser nicht unterstützten Formatelemente im konvertierten Code zu emulieren.

Das Datetime-Formatmodell in Oracle enthält mehr Elemente als Zeichenketten im Datetime-Format in Amazon Redshift. Wenn Ihr Quellcode nur Elemente im Datetime-Format enthält, die Amazon Redshift unterstützt, benötigen Sie die Funktionen des Erweiterungspakets im konvertierten Code nicht. Um zu vermeiden, dass die Funktionen des Erweiterungspakets im konvertierten Code verwendet werden, wählen Sie Datetype-Formatelemente aus, die Sie im Oracle-Code verwenden, ähneln den Zeichenketten im Datetime-Format in Amazon Redshift. In diesem Fall funktioniert der konvertierte Code schneller.

Das numerische Formatmodell in Oracle umfasst mehr Elemente als numerische Formatzeichenfolgen in Amazon Redshift. Wenn Ihr Quellcode nur numerische Formatelemente enthält, die Amazon Redshift unterstützt, benötigen Sie die Funktionen des Erweiterungspakets im konvertierten Code nicht. Um zu vermeiden, dass die Funktionen des Erweiterungspakets im konvertierten Code verwendet werden, wählen Sie Numerische Formatelemente aus, die Sie im Oracle-Code verwenden, ähneln numerischen Formatzeichenfolgen in Amazon Redshift. In diesem Fall funktioniert der konvertierte Code schneller.

- Um Oracle LEAD - und LAG Analysefunktionen zu konvertieren. Löst standardmäßig für jede LEAD LAG ND-Funktion ein Aktionselement aus. AWS SCT

Wenn Ihr Quellcode die Standardwerte für Offset in diesen Funktionen nicht verwendet, AWS SCT können Sie die Verwendung dieser Funktionen mit der NVL Funktion emulieren. Wählen Sie dazu die Funktion NVL verwenden, um das Verhalten der LEAD- und LAG-Funktionen von Oracle zu emulieren.

- Um das Verhalten von Primär- und Einzelschlüsseln in Ihrem Amazon Redshift-Cluster zu emulieren, wählen Sie das Verhalten von Primär- und Einzelschlüsseln emulieren aus.

Amazon Redshift erzwingt keine eindeutigen und primären Schlüssel und verwendet sie nur zu Informationszwecken. Wenn Sie diese Einschränkungen in Ihrem Code verwenden, stellen Sie sicher, dass ihr Verhalten im konvertierten Code AWS SCT emuliert wird.

- Um die Komprimierung auf Amazon Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu die Option Kompressionskodierung verwenden aus.

AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift-Algorithmus automatisch eine Kompressionskodierung zu. Weitere Informationen finden Sie unter [Kompressionskodierungen](#) im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Kompressionskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Kompressionskodierung verwenden ausgewählt haben.

Einstellungen für die Konvertierungsoptimierung von Oracle Data Warehouse zu Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von Oracle Data Warehouse zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Oracle und dann Oracle — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt die Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Oracle Data Warehouse zu Amazon Redshift an.

Die Einstellungen für die Konvertierungsoptimierung von Oracle Data Warehouse nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu die automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

Die automatische Tabellenoptimierung ist ein Selbstoptimierungsprozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter [Arbeiten mit automatischer Tabellenoptimierung](#) im Amazon Redshift Database Developer Guide.

Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl aus.

- Um die Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

Sie können mithilfe von Amazon Redshift-Metadaten, statistischen Informationen oder diesen beiden Optionen Sortier- und Verteilungsschlüssel auswählen. Wählen Sie auf der Registerkarte Optimierungsstrategien für die erste Schlüsselauswahlstrategie eine der folgenden Optionen aus:

- Metadaten verwenden, statistische Informationen ignorieren
- Metadaten ignorieren, statistische Informationen verwenden
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. AWS SCT definiert anhand dieser Gewichtungswerte, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie Kleine Tabellen finden definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

Sie können nicht nur das Gewicht für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung.

- Geben Sie im Feld Sortierschlüsselspaltenlimit die maximale Anzahl von Spalten in den Sortierschlüssel ein.
- Geben Sie für schiefen Schwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, deren Verzerrungswert größer als der Schwellenwert ist,

aus der Liste der Kandidaten für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des häufigsten Werts zur Gesamtzahl der Datensätze.

- Geben Sie für die Top-N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Auf der Registerkarte Optimierungsstrategien können Sie auch die Größe kleiner Tabellen für die Strategie Kleine Tabellen suchen definieren. Geben Sie für Minimale Tabellenzeilenanzahl und Maximale Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verwendung von Snowflake als Quelle für AWS SCT

Sie können AWS SCT verwenden, um Schemas, Codeobjekte und Anwendungscode von Snowflake nach Amazon Redshift zu konvertieren.

Rechte für Snowflake als Quelldatenbank

Sie können eine Rolle mit Rechten erstellen und dieser Rolle den Namen eines Benutzers zuweisen, indem Sie die SECURITYADMIN Rolle und den SECURITYADMIN Sitzungskontext verwenden.

Das folgende Beispiel erstellt minimale Rechte und gewährt sie dem `min_privs` Benutzer.

```
create role role_name;  
grant role role_name to role sysadmin;  
grant usage on database db_name to role role_name;  
grant usage on schema db_name.schema_name to role role_name;  
grant usage on warehouse datawarehouse_name to role role_name;  
grant monitor on database db_name to role role_name;  
grant monitor on warehouse datawarehouse_name to role role_name;  
grant select on all tables in schema db_name.schema_name to role role_name;  
grant select on future tables in schema db_name.schema_name to role role_name;  
grant select on all views in schema db_name.schema_name to role role_name;  
grant select on future views in schema db_name.schema_name to role role_name;  
grant select on all external tables in schema db_name.schema_name to role role_name;  
grant select on future external tables in schema db_name.schema_name to role role_name;  
grant usage on all sequences in schema db_name.schema_name to role role_name;
```

```
grant usage on future sequences in schema db_name.schema_name to role role_name;  
grant usage on all functions in schema db_name.schema_name to role role_name;  
grant usage on future functions in schema db_name.schema_name to role role_name;  
grant usage on all procedures in schema db_name.schema_name to role role_name;  
grant usage on future procedures in schema db_name.schema_name to role role_name;  
create user min_privs password='real_user_password'  
DEFAULT_ROLE = role_name DEFAULT_WAREHOUSE = 'datawarehouse_name';  
grant role role_name to user min_privs;
```

Ersetzen Sie die Platzhalter im vorherigen Beispiel wie folgt:

- Ersetzen Sie *role_name* durch den Namen einer Rolle mit schreibgeschützten Rechten.
- *db_name* Ersetzen Sie durch den Namen der Quelldatenbank.
- *schema_name* Ersetzen Sie durch den Namen des Quellschemas.
- *datawarehouse_name* Ersetzen Sie es durch den Namen eines erforderlichen Data Warehouse.
- *min_privs* Ersetzen Sie es durch den Namen eines Benutzers mit minimalen Rechten.

Die DEFAULT_WAREHOUSE Parameter DEFAULT_ROLE und sind schlüsselsensitiv.

Konfiguration des sicheren Zugriffs auf Amazon S3

Sicherheits- und Zugriffsverwaltungsrichtlinien für einen Amazon S3-Bucket ermöglichen es Snowflake, auf Daten zuzugreifen, Daten aus dem S3-Bucket zu lesen und in diesen zu schreiben. Sie können den sicheren Zugriff auf einen privaten Amazon S3-Bucket mithilfe des STORAGE INTEGRATION Snowflake-Objektyps konfigurieren. Ein Snowflake-Speicherintegrationsobjekt delegiert die Authentifizierungsverantwortung an eine Snowflake-Identitäts- und Zugriffsverwaltungseinheit.

Weitere Informationen finden Sie unter [Konfiguration einer Snowflake-Speicherintegration für den Zugriff auf Amazon S3](#) in der Snowflake-Dokumentation.

Verbindung zu Snowflake als Quelle herstellen

Gehen Sie wie folgt vor, um mit dem eine Verbindung zu Ihrer Quelldatenbank herzustellen AWS Schema Conversion Tool.

So stellen Sie eine Verbindung zu einer Snowflake-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.

2. Wählen Sie Snowflake und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankanmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:

- Gehen Sie wie folgt vor, um die Datenbankanmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankanmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen für das Snowflake-Quell-Data Warehouse manuell einzugeben:

Parameter	Action
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Snowflake-Datenbank ein.
User name und Password	Geben Sie die Datenbankanmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen. AWS SCT speichert Ihr Passwort nur dann in einem verschlüsselten Format, wenn Sie es ausdrücklich anfordern.

Parameter	Action
Use SSL	<p>Wählen Sie diese Option, wenn Sie Secure Sockets Layer (SSL) verwenden möchten, um eine Verbindung zu Ihrer Datenbank herzustellen. Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an:</p> <ul style="list-style-type: none"> • Pfad des privaten Schlüssels: Der Speicherort eines privaten Schlüssels. • Passphrase: Die Passphrase für den privaten Schlüssel. <p>Weitere Informationen zur SSL-Unterstützung für Snowflake finden Sie unter Konfigurieren von Sicherheitsoptionen für Verbindungen.</p>
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option festlegen, können Sie das Datenbankkennwort speichern. Auf diese Weise können Sie schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.</p>
Snowflake-Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projektinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einschränkungen für Snowflake als Quelle

Die folgenden Einschränkungen gelten bei der Verwendung von Snowflake als Quelle für: AWS SCT

- Objektkennungen müssen im Kontext des Objekttyps und des übergeordneten Objekts eindeutig sein:

Datenbank

Schema-IDs müssen innerhalb einer Datenbank eindeutig sein.

Schemata

Objektbezeichner, z. B. für Tabellen und Ansichten, müssen innerhalb eines Schemas eindeutig sein.

Tabellen/Ansichten

Spaltenbezeichner müssen innerhalb einer Tabelle eindeutig sein.

- Die maximale Anzahl von Tabellen für große und extragroße Cluster-Knotentypen beträgt 9.900. Für 8 große Cluster-Knotentypen beträgt die maximale Anzahl von Tabellen 100.000. Das Limit umfasst temporäre Tabellen, die sowohl benutzerdefiniert als auch von Amazon Redshift während der Abfrageverarbeitung oder Systemwartung erstellt wurden. Weitere Informationen finden Sie unter [Amazon Redshift quotas](#) (Amazon-Redshift-Kontingente) im Amazon-Redshift-Clusterverwaltungshandbuch.
- Für gespeicherte Prozeduren beträgt die maximale Anzahl von Eingabe- und Ausgabeargumenten 32.

Quelldatentypen für Snowflake

Im Folgenden finden Sie die Snowflake-Quelldatentypen, die bei der Verwendung unterstützt werden, AWS SCT und die Standardzuordnung zu einem Amazon Redshift-Ziel.

Snowflake-Datentypen	Amazon Redshift-Datentypen
NUMBER	NUMERISCH (38)
NUMBER(p)	Wenn $p \leq 4$ ist, dann SMALLINT Wenn $p \geq 5$ und ≤ 9 ist, dann INTEGER

Snowflake-Datentypen	Amazon Redshift-Datentypen
	<p>Wenn $p \Rightarrow 10$ und ≤ 18 ist, dann BIGINT</p> <p>Wenn $p \Rightarrow 19$ ist, dann NUMERIC (p)</p>
ZAHL (p, 0)	<p>Wenn $p \leq 4$ ist, dann SMALLINT</p> <p>Wenn $p \Rightarrow 5$ und ≤ 9 ist, dann INTEGER</p> <p>Wenn $p \Rightarrow 10$ und ≤ 18 ist, dann BIGINT</p> <p>Wenn $p \Rightarrow 19$ ist, dann: NUMERISCH (p,0)</p>
ZAHL (p, s)	<p>Wenn $p \Rightarrow 1$ und ≤ 38 ist und wenn $s \Rightarrow 1$ und ≤ 37 ist, dann</p> <p>NUMERIC (p,s)</p>
FLOAT	FLOAT
<p>TEXT</p> <p>Unicode-Zeichen bis zu 16.777.216 Byte; bis zu 4 Byte pro Zeichen.</p>	VARCHAR (MAXIMAL)
<p>TEXT (p)</p> <p>Unicode-Zeichen bis zu 65.535 Byte; bis zu 4 Byte pro Zeichen.</p>	Wenn $p \leq 65.535$ ist, dann ist VARCHAR (p)
<p>TEXT (p)</p> <p>Unicode-Zeichen bis zu 16.777.216 Byte; bis zu 4 Byte pro Zeichen.</p>	Wenn $p \Rightarrow 65.535$ ist und $\leq 16.777.216$ dann, VARCHAR (MAX)
<p>BINARY</p> <p>Einzelbyte-Zeichen bis zu 8.388.608 Byte; 1 Byte pro Zeichen.</p>	VARCHAR (MAXIMAL)

Snowflake-Datentypen	Amazon Redshift-Datentypen
BINÄR (p) Einzelbyte-Zeichen bis zu 65.535 Byte; 1 Byte pro Zeichen.	VARCHAR (p)
BINÄR (p) Einzelbyte-Zeichen bis zu 8.388.608 Byte; 1 Byte pro Zeichen.	VARCHAR (MAXIMAL)
BOOLEAN	BOOLEAN
DATUM	DATUM
TIME Zeitwerte zwischen 00:00:00 und 23:59:59.999 999999.	VARCHAR (18)
ZEIT (f) Zeitwerte zwischen 00:00:00 und 23:59:59:59 (f).	VARCHAR (n) — 9 + dt-attr-1
TIMESTAMP_NTZ	TIMESTAMP
TIMESTAMP_TZ	TIMESTAMPTZ

Einstellungen für die Konvertierung von Snowflake nach Amazon Redshift

Um die Konvertierungseinstellungen von Snowflake in Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Snowflake und dann Snowflake — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Snowflake nach Amazon Redshift an.

Die Konvertierungseinstellungen von Snowflake nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift-Zielcluster gelten AWS SCT können.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift-Cluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendung von Tabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt es je nach Knotentyp die Anzahl der Tabellen, die auf Ihren Amazon Redshift-Ziel-Cluster angewendet werden sollen. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter [Kontingente und Limits in Amazon Redshift](#) im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert alle Ihre Quelltabellen, auch wenn dies mehr ist, als Ihr Amazon Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie bei der Anwendung des konvertierten Codes das Amazon Redshift-Cluster-Kontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT wendet Tabellen außerdem auf Ihren Amazon Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um die Komprimierung auf Amazon Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu die Option Kompressionskodierung verwenden aus.

AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift-Algorithmus automatisch eine Kompressionskodierung zu. Weitere Informationen finden Sie unter [Kompressionskodierungen](#) im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Kompressionskodierung für KEY-

Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Kompressionskodierung verwenden ausgewählt haben.

Einstellungen für die Konvertierungsoptimierung von Snowflake nach Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von Snowflake in Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen.

Wählen Sie in der oberen Liste Snowflake und dann Snowflake — Amazon Redshift aus.

Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt die Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Snowflake nach Amazon Redshift an.

Die Einstellungen für die Konversionsoptimierung von Snowflake nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu die automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

Die automatische Tabellenoptimierung ist ein Selbstoptimierungsprozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter [Arbeiten mit automatischer Tabellenoptimierung](#) im Amazon Redshift Database Developer Guide.

Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl aus.

- Um die Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

Sie können mithilfe von Amazon Redshift-Metadaten, statistischen Informationen oder diesen beiden Optionen Sortier- und Verteilungsschlüssel auswählen. Wählen Sie auf der Registerkarte Optimierungsstrategien für die erste Schlüsselauswahlstrategie eine der folgenden Optionen aus:

- Metadaten verwenden, statistische Informationen ignorieren
- Metadaten ignorieren, statistische Informationen verwenden
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. AWS SCT definiert anhand dieser Gewichtungswerte, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie Kleine Tabellen finden definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCTwendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

Sie können nicht nur das Gewicht für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung.

- Geben Sie im Feld Sortierschlüsselspaltenlimit die maximale Anzahl von Spalten in den Sortierschlüssel ein.
- Geben Sie für schiefen Schwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCTschließt Spalten, deren Verzerrungswert größer als der Schwellenwert ist, aus der Liste der Kandidaten für den Verteilungsschlüssel aus. AWS SCTdefiniert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des häufigsten Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top-N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Auf der Registerkarte Optimierungsstrategien können Sie auch die Größe kleiner Tabellen für die Strategie Kleine Tabellen suchen definieren. Geben Sie für Minimale Tabellenzeilenanzahl und Maximale Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCTwendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verwendung von Microsoft SQL Server Data Warehouse als Quelle für AWS SCT

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Microsoft SQL Server DW in Amazon Redshift oder Amazon Redshift zu konvertieren und in Kombination zu AWS Glue verwenden.

Rechte für Microsoft SQL Server Data Warehouse als Quelle

Die folgenden Rechte sind erforderlich, um das Microsoft SQL Server Data Warehouse als Quelle verwenden zu können:

- VIEW DEFINITION
- VIEW DATABASE STATE
- SELECT ON SCHEMA :: *<schema_name>*

Ersetzen Sie im vorherigen Beispiel den `<source_schema>` Platzhalter durch den Namen des Quelltextschemas.

Gewähren Sie die erforderlichen Berechtigungen für alle Datenbanken, deren Schemata Sie konvertieren.

Gewähren Sie darüber hinaus folgende Berechtigungen für die Master-Datenbank:

- VIEW SERVER STATE

Einschränkungen für SQL Server Data Warehouse als Quelle

Die Verwendung von Microsoft SQL Server Parallel Data Warehouse (PDW) als Quelle wird derzeit nicht unterstützt.

Verbindung zu SQL Server Data Warehouse als Quelle herstellen

Gehen Sie wie folgt vor, um mit dem eine Verbindung zu Ihrer SQL Server Data Warehouse-Quelldatenbank herzustellenAWS Schema Conversion Tool.

So stellen Sie eine Verbindung zu einer SQL Server Data Warehouse-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Microsoft SQL Server und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankanmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:

- Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Microsoft SQL Server-Quell-Data Warehouse-Verbindungsinformationen manuell einzugeben:

Parameter	Action
Server name	Geben Sie den Domain Name Service (DNS)-Namen oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Instance name	Geben Sie den Instanznamen für das SQL Server Data Warehouse ein.
User name und Password	<p>Geben Sie die Datenbankmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>

Parameter	Action
Use SSL	<p>Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden . Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an:</p> <ul style="list-style-type: none"> • Serverzertifikat vertrauen: Wählen Sie diese Option, um dem Serverzertifikat zu vertrauen. • Trust Store: Ein Trust Store, den Sie in den globalen Einstellungen einrichten.
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbank kennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.</p>
Pfad zum SQL Server-Treiber	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projekteinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden AWS SCT kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einstellungen für die Konvertierung von SQL Server Data Warehouse nach Amazon Redshift

Um die Konvertierungseinstellungen von SQL Server Data Warehouse zu Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen

Sie in der oberen Liste Microsoft SQL Server und dann Microsoft SQL Server — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von SQL Server Data Warehouse zu Amazon Redshift an.

Die Konvertierungseinstellungen von SQL Server Data Warehouse nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift-Zielcluster gelten AWS SCT können.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift-Cluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendung von Tabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt es je nach Knotentyp die Anzahl der Tabellen, die auf Ihren Amazon Redshift-Ziel-Cluster angewendet werden sollen. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter [Kontingente und Limits in Amazon Redshift](#) im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert alle Ihre Quelltabellen, auch wenn dies mehr ist, als Ihr Amazon Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie bei der Anwendung des konvertierten Codes das Amazon Redshift-Cluster-Kontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT wendet Tabellen außerdem auf Ihren Amazon Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um Partitionen der Quelltable in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltable erstellt werden AWS SCT können.

Amazon Redshift unterstützt keine Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltable in eine separate Tabelle in Amazon Redshift migrieren. Erstellen Sie AWS SCT dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

AWS SCTbestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltable. Je nach Art der Partitionierung der Quelltable kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon Redshift-Cluster anwenden können. Um zu vermeiden, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltable erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage eines Jahres und zwei Tabellen für NO RANGE und UNKNOWN Partitionen entspricht.

- Um die Komprimierung auf Amazon Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu die Option Kompressionskodierung verwenden aus.

AWS SCTweist Spalten mithilfe des standardmäßigen Amazon Redshift-Algorithmus automatisch eine Kompressionskodierung zu. Weitere Informationen finden Sie unter [Kompressionskodierungen](#) im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Kompressionskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Kompressionskodierung verwenden ausgewählt haben.

Einstellungen für die Konvertierungsoptimierung von SQL Server Data Warehouse zu Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von SQL Server Data Warehouse in Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Microsoft SQL Server und dann Microsoft SQL Server — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCTzeigt die Einstellungen zur Konvertierungsoptimierung für die Konvertierung von SQL Server Data Warehouse in Amazon Redshift an.

Die Einstellungen für die Konvertierungsoptimierung von SQL Server Data Warehouse nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu die automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

Die automatische Tabellenoptimierung ist ein Selbstoptimierungsprozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter [Arbeiten mit automatischer Tabellenoptimierung](#) im Amazon Redshift Database Developer Guide.

Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl aus.

- Um die Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

Sie können mithilfe von Amazon Redshift-Metadaten, statistischen Informationen oder diesen beiden Optionen Sortier- und Verteilungsschlüssel auswählen. Wählen Sie auf der Registerkarte Optimierungsstrategien für die erste Schlüsselauswahlstrategie eine der folgenden Optionen aus:

- Metadaten verwenden, statistische Informationen ignorieren
- Metadaten ignorieren, statistische Informationen verwenden
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. AWS SCT definiert anhand dieser Gewichtungswerte, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie Kleine Tabellen finden definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

Sie können nicht nur das Gewicht für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung.

- Geben Sie im Feld Sortierschlüsselspaltenlimit die maximale Anzahl von Spalten in den Sortierschlüssel ein.
- Geben Sie für schiefen Schwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCT schließt Spalten, deren Verzerrungswert größer als der Schwellenwert ist,

aus der Liste der Kandidaten für den Verteilungsschlüssel aus. AWS SCT definiert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des häufigsten Werts zur Gesamtzahl der Datensätze.

- Geben Sie für die Top-N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Auf der Registerkarte Optimierungsstrategien können Sie auch die Größe kleiner Tabellen für die Strategie Kleine Tabellen suchen definieren. Geben Sie für Minimale Tabellenzeilenanzahl und Maximale Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verwendung von Teradata als Quelle für AWS SCT

Sie können AWS SCT verwenden, um Schemas, Codeobjekte und Anwendungscode von Teradata nach Amazon Redshift oder Amazon Redshift zu konvertieren und AWS Glue in Kombination zu verwenden.

Rechte für Teradata als Quelle

Für die Verwendung von Teradata als Quelle sind die folgenden Rechte erforderlich:

- SELECT ON DBC
- WÄHLE AUF SYSUDTLIB
- AUF SYSLIB AUSWÄHLEN
- WÄHLEN SIE AUF <source_database>
- PROZEDUR ERSTELLEN AM <source_database>

Ersetzen Sie im vorherigen Beispiel den <source_database>Platzhalter durch den Namen der Quelldatenbank.

AWS SCT benötigt das CREATE PROCEDURE-Privileg, um HELP PROCEDURE für alle Prozeduren in der Quelldatenbank auszuführen. AWS SCT verwendet dieses Privileg nicht, um neue Objekte in Ihrer Teradata-Quelldatenbank zu erstellen.

Verbindung zu Teradata als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Teradata-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Teradata-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Teradata und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankanmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um die Datenbankanmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankanmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen der Teradata-Quelldatenbank manuell einzugeben:

Parameter	Action
Verbindungsname	Geben Sie einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.

Parameter	Action
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Teradata-Datenbank ein.
User name und Password	<p>Geben Sie die Datenbankanmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>
Passwort speichern	AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankkennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.
Daten verschlüsseln	Wählen Sie diese Option, um Daten zu verschlüsseln, die Sie mit der Datenbank austauschen. Wenn Sie diese Option wählen, wird die Portnummer 443 verwendet, um verschlüsselte Daten zwischen AWS SCT und Ihrer Teradata-Datenbank zu übertragen.

Parameter	Action
Teradata-Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projekteinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

- Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden kann.
- Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

LDAP-Authentifizierung mit einer Teradata-Quelle verwenden

Gehen Sie wie folgt vor, um eine Lightweight Directory Access Protocol (LDAP)-Authentifizierung für Teradata-Benutzer einzurichten, die Microsoft Active Directory in Windows verwenden.

Im folgenden Verfahren ist die Active Directory-Domäne `test.local.com`. Der Windows-Server ist DC, und er ist mit Standardeinstellungen konfiguriert. Das folgende Skript erstellt das `test_ldap` Active Directory-Konto, und dieses Konto verwendet das `test_ldap` Passwort.

So richten Sie die LDAP-Authentifizierung für Teradata-Benutzer ein, die Microsoft Active Directory in Windows ausführen

- Bearbeiten Sie im Verzeichnis `/opt/teradata/tdat/tdgss/site` die Datei `TdgssUserConfigFile.xml`. Ändern Sie den LDAP-Abschnitt wie folgt.

```
AuthorizationSupported="no"

LdapServerName="DC.test.local.com"
LdapServerPort="389"
LdapServerRealm="test.local.com"
LdapSystemFQDN="dc= test, dc= local, dc=com"
LdapBaseFQDN="dc=test, dc=local, dc=com"
```

- Übernehmen Sie die Änderungen, indem Sie die Konfiguration wie folgt ausführen.

```
#cd /opt/teradata/tdgss/bin
#./run_tdgssconfig
```

3. Testen Sie die Konfiguration, indem Sie den folgenden Befehl ausführen.

```
# /opt/teradata/tdat/tdgss/14.10.03.01/bin/tdsbind -u test_ldap -w test_ldap
```

Die Ausgabe sollte folgendermaßen oder ähnlich aussehen.

```
LdapGroupBaseFQDN: dc=Test, dc=local, dc=com
LdapUserBaseFQDN: dc=Test, dc=local, dc=com
LdapSystemFQDN: dc= test, dc= local, dc=com
LdapServerName: DC.test.local.com
LdapServerPort: 389
LdapServerRealm: test.local.com
LdapClientUseTls: no
LdapClientTlsReqCert: never
LdapClientMechanism: SASL/DIGEST-MD5
LdapServiceBindRequired: no
LdapClientTlsCRLCheck: none
LdapAllowUnsafeServerConnect: yes
UseLdapConfig: no
AuthorizationSupported: no
FQDN: CN=test, CN=Users, DC=Anthem, DC=local, DC=com
AuthUser: ldap://DC.test.local.com:389/CN=test1,CN=Users,DC=test,DC=local,DC=com
DatabaseName: test
Service: tdsbind
```

4. Starten Sie TPA mit dem folgenden Befehl neu.

```
#tpareset -f "use updated TDGSSCONFIG GD0"
```

5. Erstellen Sie, wie im Folgenden gezeigt, denselben Benutzer in der Teradata-Datenbank wie in Active Directory.

```
CREATE USER test_ldap AS PERM=1000, PASSWORD=test_ldap;
GRANT LOGON ON ALL TO test WITH NULL PASSWORD;
```

Wenn Sie das Benutzerkennwort in Active Directory für Ihren LDAP-Benutzer ändern, geben Sie dieses neue Passwort während der Verbindung zu Teradata im LDAP-Modus an. Im DEFAULT-

Modus stellen Sie eine Verbindung zu Teradata her, indem Sie den LDAP-Benutzernamen und ein beliebiges Passwort verwenden.

Konfiguration der Statistikerfassung in Ihrem Teradata-Quell-Data Warehouse

Um Ihr Teradata-Quell-Data Warehouse zu konvertieren, AWS SCT verwendet es Statistiken, um Ihr konvertiertes Amazon Redshift-Data Warehouse zu optimieren. Sie können Statistiken in der Statistikdatei sammeln AWS SCT oder hochladen. Weitere Informationen finden Sie unter [Statistiken sammeln oder hochladen](#).

Führen Sie die folgenden erforderlichen Aufgaben aus, um sicherzustellen, dass Statistiken aus Ihrem Data Warehouse erfasst werden AWS SCT können.

Um Statistiken aus Ihrem Teradata Data Warehouse zu sammeln

1. Führen Sie die folgende Abfrage aus, um Statistiken für alle Tabellen in Ihrem Data Warehouse abzurufen.

```
collect summary statistics on table_name;
```

Ersetzen Sie im vorherigen Beispiel *table_name* durch den Namen Ihrer Quelltable. Wiederholen Sie die Abfrage für jede Tabelle, die Sie konvertieren.

2. Führen Sie die folgende Abfrage aus, um die Kontozeichenfolge für den Benutzer zu ermitteln, die Sie für die Konvertierung Ihres Data Warehouse verwenden.

```
select * from dbc.accountinfo where username = 'user_name'
```

3. Aktivieren Sie die Abfrageprotokollierung für einen bestimmten Benutzer mithilfe der Kontozeichenfolge aus dem vorherigen Beispiel.

```
BEGIN QUERY LOGGING WITH OBJECTS, SQL ON ALL ACCOUNT=('M$BUSI$$D$H');
```

Alternativ können Sie die Abfrageprotokollierung für alle Datenbankbenutzer aktivieren.

```
BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT SQLTEXT=0 ON ALL;
```

Nachdem Sie die Erfassung der Data Warehouse-Statistiken abgeschlossen haben, deaktivieren Sie die Abfrageprotokollierung. Dazu können Sie das folgende Codebeispiel verwenden.


```
end query logging with explain, objects, sql on all account=(' $M$BUSI$$D$H');
```

Erfassung von Statistiken im Offline-Modus aus Ihrem Teradata-Quell-Data Warehouse

Nachdem Sie die Statistikerfassung in Ihrem Teradata Data Warehouse konfiguriert haben, können Sie Statistiken in Ihrem AWS SCT Projekt sammeln. Alternativ können Sie Basic Teradata Query (BTEQ) -Skripts verwenden, um Statistiken im Offline-Modus zu sammeln. Anschließend können Sie die Dateien mit den gesammelten Statistiken in Ihr AWS SCT Projekt hochladen. Weitere Informationen finden Sie unter [Statistiken sammeln oder hochladen](#).

Um Statistiken aus Ihrem Teradata Data Warehouse im Offline-Modus zu sammeln

1. Erstellen Sie das `off-line_stats.bteq` Skript mit dem folgenden Inhalt.

```
.OS IF EXIST column-stats-tera.csv del /F column-stats-tera.csv
.OS IF EXIST table-stats-tera.csv del /F table-stats-tera.csv
.OS IF EXIST column-skew-script-tera.csv del /F column-skew-script-tera.csv
.OS IF EXIST column-skew-stats-tera.csv del /F column-skew-stats-tera.csv
.OS IF EXIST query-stats-tera.csv del /F query-stats-tera.csv
.LOGON your_teradata_server/your_login, your_password
.EXPORT REPORT FILE = table-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

SELECT
  ''' || OREPLACE(COALESCE(c.DatabaseName, ''), '', '""') || ';' ||
  ''' || OREPLACE(COALESCE(c.TableName, ''), '', '""') || ';' ||
  ''' || TRIM(COALESCE(s.reference_count, '0')) || ';' ||
  ''' || TRIM(COALESCE(CAST(p.RowCount AS BIGINT), '0')) || ';' ||
  ''' || CAST(CAST(w.size_in_mb AS DECIMAL (38,1) FORMAT 'Z9.9') AS VARCHAR(38))
  || ';' ||
  ''' || TRIM(COALESCE(r.stat_fk_dep_count, '0')) || ';' ||
  ''' || CAST(CAST(current_timestamp(0) as timestamp(0) format 'YYYY-MM-
DDBHH:MI:SS') as VARCHAR(19)) || '''
(TITLE
  "database_name";"table_name";"reference_count";"row_count";"size_in_mb";"stat_fk_dep_coun
FROM (select databasename, tablename
      from DBC.tablesv
      where tablekind IN ('T','0')
      and databasename = 'your_database_name'
```

```

        ) c
left join
    (select DatabaseName, TableName, max(RowCount) RowCount
     from dbc.tableStatsv
     group by 1,2)p
on p.databasename = c.databasename
and p.tablename = c.tablename
left join
    (SELECT r.ChildDB as DatabaseName,
     r.ChildTable as TableName,
     COUNT(DISTINCT r.ParentTable) reference_count
     FROM DBC.All_RI_ChildrenV r
     GROUP BY r.ChildDB, r.ChildTable) s
on s.databasename = c.databasename
and s.tablename = c.tablename
left join
    (SELECT r.ParentDB as DatabaseName,
     r.ParentTable as TableName,
     COUNT(DISTINCT r.ChildTable) stat_fk_dep_count
     FROM DBC.All_RI_ParentsV r
     GROUP BY r.ParentDB, r.ParentTable) r
on r.databasename = c.databasename
and r.tablename = c.tablename
left join
    (select databasename, tablename,
     sum(currentperm)/1024/1024 as size_in_mb
     from dbc.TableSizeV
     group by 1,2) w
on w.databasename = c.databasename
and w.tablename = c.tablename
WHERE COALESCE(r.stat_fk_dep_count,0) + COALESCE(CAST(p.RowCount AS BIGINT),0) +
     COALESCE(s.reference_count,0) > 0;

.EXPORT RESET

.EXPORT REPORT FILE = column-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000
    ''' || TRIM(COALESCE(CAST(t2.card AS BIGINT), '0')) || ';' ||

SELECT
    ''' || OREPLACE(COALESCE(trim(tv.DatabaseName), ''), '', '""') || ';' ||
    ''' || OREPLACE(COALESCE(trim(tv.TableName), ''), '', '""') || ';' ||
    ''' || OREPLACE(COALESCE(trim(tv.columnname), ''), '', '""') || ';' ||

```

```

''' || TRIM(COALESCE(CAST(t2.card AS BIGINT), '0')) ||
';' ||

''' || CAST(current_timestamp AS VARCHAR(19)) || ''' (TITLE
"database_name";"table_name";"column_name";"cardinality";"current_ts")
FROM dbc.columnsv tv
LEFT JOIN
(
SELECT
  c.DatabaseName AS DATABASE_NAME,
  c.TABLENAME AS TABLE_NAME,
  c.ColumnName AS COLUMN_NAME,
  c.UniqueValueCount AS CARD
FROM dbc.tablestatsv c
WHERE c.DatabaseName = 'your_database_name'
AND c.RowCount <> 0
) t2
ON tv.DATABASENAME = t2.DATABASE_NAME
AND tv.TABLENAME = t2.TABLE_NAME
AND tv.COLUMNNAME = t2.COLUMN_NAME
WHERE t2.card > 0;

.EXPORT RESET

.EXPORT REPORT FILE = column-skew-script-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

SELECT
'SELECT CAST(''' || TRIM(c.DatabaseName) || ''';'' || TRIM(c.TABLENAME) || ''';''
|| TRIM(c.COLUMNNAME) || ''';'' ||
TRIM(CAST(COALESCE(MAX(cnt) * 1.0 / SUM(cnt), 0) AS NUMBER FORMAT '9.9999')) ||
''';'' ||
CAST(CURRENT_TIMESTAMP(0) AS VARCHAR(19)) || '''' AS VARCHAR(512))
AS """"DATABASE_NAME""";""TABLE_NAME"";""COLUMN_NAME"";""SKEWED"";""CURRENT_TS""""
FROM(
SELECT COUNT(*) AS cnt
FROM ''' || c.DATABASENAME || '''.''' || c.TABLENAME ||
''' GROUP BY ''' || c.COLUMNNAME || ''') t' ||
CASE WHEN ROW_NUMBER() OVER(PARTITION BY c.DATABASENAME
ORDER BY c.TABLENAME DESC, c.COLUMNNAME DESC) <> 1
THEN ' UNION ALL '
ELSE ';' END (TITLE '--SKEWED--')
FROM dbc.columnsv c

```

```

INNER JOIN
(SELECT databasename, TABLENAME
FROM dbc.tablesv WHERE tablekind = 'T'
AND databasename = 'your_database_name') t
ON t.databasename = c.databasename
AND t.TABLENAME = c.TABLENAME
INNER JOIN
(SELECT databasename, TABLENAME, columnname FROM dbc.indices GROUP BY 1,2,3
WHERE TRANSLATE_CHK (databasename USING LATIN_TO_UNICODE) + TRANSLATE_CHK
(TABLENAME USING LATIN_TO_UNICODE) + TRANSLATE_CHK (columnname USING
LATIN_TO_UNICODE) = 0
) i
ON i.databasename = c.databasename
AND i.TABLENAME = c.TABLENAME
AND i.columnname = c.columnname
WHERE c.ColumnType NOT IN ('CO', 'JN', 'N', '++', 'VA', 'UT', 'AN', 'XM', 'A1', 'B0')
ORDER BY c.TABLENAME, c.COLUMNNAME;

.EXPORT RESET

.EXPORT REPORT FILE = column-skew-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

.RUN FILE = column-skew-script-tera.csv

.EXPORT RESET

.EXPORT REPORT FILE = query-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 32000

SELECT
  ''' || RTRIM(CAST(SqlTextInfo AS VARCHAR(31900))), ';' || ';' ||
  TRIM(QueryCount) || ';' ||
  TRIM(QueryId) || ';' ||
  TRIM(SqlRowNo) || ';' ||
  TRIM(QueryParts) || ';' ||
  CAST(CURRENT_TIMESTAMP(0) AS VARCHAR(19)) || ''
(TITLE
  "query_text";"query_count";"query_id";"sql_row_no";"query_parts";"current_ts")
FROM
  (
    SELECT QueryId, SqlTextInfo, SqlRowNo, QueryParts, QueryCount,

```

```

SUM(QueryFirstRow) OVER (ORDER BY QueryCount DESC, QueryId ASC, SqlRowNo ASC
ROWS UNBOUNDED PRECEDING) AS topN
FROM
(SELECT QueryId,  SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
CASE WHEN
ROW_NUMBER() OVER (PARTITION BY QueryCount, SqlTextInfo ORDER BY QueryId,
SqlRowNo) = 1 AND SqlRowNo = 1
THEN 1 ELSE 0 END AS QueryFirstRow
FROM (
SELECT q.QueryId,  q.SqlTextInfo, q.SqlRowNo,
MAX(q.SqlRowNo) OVER (PARTITION BY q.QueryId) QueryParts,
COUNT(q.SqlTextInfo) OVER (PARTITION BY q.SqlTextInfo) QueryCount
FROM DBC.dbqlsqltbl q
INNER JOIN
(
SELECT QueryId
FROM DBC.DBQLogTbl t
WHERE TRIM(t.StatementType) IN ('SELECT')
AND TRIM(t.AbortFlag) = '' AND t.ERRORCODE = 0
AND (CASE WHEN 'All users' IN ('All users') THEN 'All users' ELSE
TRIM(t.USERNAME) END) IN ('All users') --user_name list
AND t.StartTime > CURRENT_TIMESTAMP - INTERVAL '30' DAY
GROUP BY 1
) t
ON q.QueryId = t.QueryId
INNER JOIN
(
SELECT QueryId
FROM DBC.QryLogObjectsV
WHERE ObjectDatabaseName = 'your_database_name'
AND ObjectType = 'Tab'
AND CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
GROUP BY 1
) r
ON r.QueryId = t.QueryId
WHERE q.CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
) t
) t
WHERE SqlTextInfo NOT LIKE '%"%"
) q
WHERE
QueryParts >=1
AND topN <= 50
ORDER BY QueryCount DESC, QueryId, SqlRowNo

```

```

QUALIFY COUNT(QueryId) OVER (PARTITION BY QueryId) = QueryParts;

.EXPORT RESET

.LOGOFF

.QUIT

```

- Erstellen Sie die `td_run_bteq.bat` Datei, die das BTEQ-Skript ausführt, das Sie im vorherigen Schritt erstellt haben. Verwenden Sie den folgenden Inhalt für diese Datei.

```

@echo off > off-line_stats1.bteq & setLocal enableDELAYedexpansion
@echo off > off-line_stats2.bteq & setLocal enableDELAYedexpansion

set old1=your_teradata_server
set new1=%1
set old2=your_login
set new2=%2
set old3=your_database_name
set new3=%3
set old4=your_password
set /p new4=Input %2 pass?

for /f "tokens=* delims=" %a in (off-line_stats.bteq) do (
set str1=%a
set str1=!str1:%old1%=%new1%!
>> off-line_stats1.bteq echo !str1!
)

for /f "tokens=* delims=" %a in (off-line_stats1.bteq) do (
set str2=%a
set str2=!str2:%old2%=%new2%!
>> off-line_stats2.bteq echo !str2!
)

type nul > off-line_stats1.bteq

for /f "tokens=* delims=" %a in (off-line_stats2.bteq) do (
set str3=%a
set str3=!str3:%old3%=%new3%!
>> off-line_stats1.bteq echo !str3!
)

```

```
type nul > off-line_stats2.bteq

for /f "tokens=* delims= " %%a in (off-line_stats1.bteq) do (
set str4=%%a
set str4=!str4:%old4%=!new4%!
>> off-line_stats2.bteq echo !str4!
)

del .\off-line_stats1.bteq

echo export starting...

bteq -c UTF8 < off-line_stats.bteq > metadata_export.log

pause
```

- Erstellen Sie die `runme.bat` Datei, in der die Batch-Datei ausgeführt wird, die Sie im vorherigen Schritt erstellt haben. Verwenden Sie den folgenden Inhalt für diese Datei.

```
.\td_run_bteq.bat ServerName UserName DatabaseName
```

Ersetzen Sie in der `runme.bat` Datei *ServerName**UserName*, und *DatabaseName* durch Ihre zutreffenden Werte.

Führen Sie dann die `runme.bat` Datei aus. Wiederholen Sie diesen Schritt für jedes Data Warehouse, das Sie zu Amazon Redshift konvertieren.

Nachdem Sie dieses Skript ausgeführt haben, erhalten Sie drei Dateien mit Statistiken für jede Datenbank. Sie können diese Dateien in Ihr AWS SCT Projekt hochladen. Wählen Sie dazu im linken Bereich Ihres Projekts Ihr Data Warehouse aus und öffnen Sie das Kontextmenü (Rechtsklick). Wählen Sie Upload Statistics aus.

Einstellungen für die Konvertierung von Teradata nach Amazon Redshift

Um die Konvertierungseinstellungen von Teradata nach Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Teradata und dann Teradata — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Teradata nach Amazon Redshift an.

Die Konvertierungseinstellungen von Teradata nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift-Zielcluster gelten AWS SCT können.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift-Cluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendung von Tabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt es je nach Knotentyp die Anzahl der Tabellen, die auf Ihren Amazon Redshift-Ziel-Cluster angewendet werden sollen. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter [Kontingente und Limits in Amazon Redshift](#) im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert alle Ihre Quelltabellen, auch wenn dies mehr ist, als Ihr Amazon Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie bei der Anwendung des konvertierten Codes das Amazon Redshift-Cluster-Kontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT wendet Tabellen außerdem auf Ihren Amazon Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um Partitionen der Quelltable in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltable erstellt werden AWS SCT können.

Amazon Redshift unterstützt keine Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltable in eine separate Tabelle in Amazon Redshift migrieren. Erstellen Sie AWS SCT dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

AWS SCT bestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltable. Je nach Art der Partitionierung der Quelltable kann diese Zahl das Kontingent für die Tabellen überschreiten,

das Sie auf Ihren Amazon Redshift-Cluster anwenden können. Um zu vermeiden, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltable erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage eines Jahres und zwei Tabellen für NO RANGE und UNKNOWN Partitionen entspricht.

- Um die Komprimierung auf Amazon Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu die Option Kompressionskodierung verwenden aus.

AWS SCT weist Spalten mithilfe des standardmäßigen Amazon Redshift-Algorithmus automatisch eine Kompressionskodierung zu. Weitere Informationen finden Sie unter [Kompressionskodierungen](#) im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Kompressionskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Kompressionskodierung verwenden ausgewählt haben.

- Um eine explizite Liste von Spalten in konvertiertem Code für SELECT * Anweisungen zu verwenden, wählen Sie Explizite Spaltendeklaration verwenden aus.
- Um das Verhalten von Primär- und Einzelschlüsseln in Ihrem Amazon Redshift-Cluster zu emulieren, wählen Sie das Verhalten von Primär- und Einzelschlüsseln emulieren aus.

Amazon Redshift erzwingt keine eindeutigen und primären Schlüssel und verwendet sie nur zu Informationszwecken. Wenn Sie diese Einschränkungen in Ihrem Code verwenden, stellen Sie sicher, dass ihr Verhalten im konvertierten Code AWS SCT emuliert wird.

- Um die Einzigartigkeit der Daten in den Amazon Redshift-Zieltabellen sicherzustellen. Wählen Sie dazu das Verhalten von SET-Tabellen emulieren aus.

Teradata erstellt Tabellen mit dem SET Syntaxelement als Standardoption. Sie können einer SET Tabelle keine doppelten Zeilen hinzufügen. Wenn Ihr Quellcode diese Eindeutigkeitsbeschränkung nicht verwendet, deaktivieren Sie diese Option. In diesem Fall funktioniert der konvertierte Code schneller.

Wenn Ihr Quellcode die SET Option in Tabellen als Eindeutigkeitsbeschränkung verwendet, aktivieren Sie diese Option. In diesem Fall werden die INSERT . . SELECT Anweisungen im konvertierten Code AWS SCT neu geschrieben, um das Verhalten Ihrer Quelldatenbank zu emulieren.

Einstellungen für die Konversionsoptimierung von Teradata nach Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von Teradata nach Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Teradata und dann Teradata — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCT zeigt die Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Teradata nach Amazon Redshift an.

Die Einstellungen für die Konversionsoptimierung von Teradata nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu die automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

Die automatische Tabellenoptimierung ist ein Selbstoptimierungsprozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter [Arbeiten mit automatischer Tabellenoptimierung](#) im Amazon Redshift Database Developer Guide.

Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl aus.

- Um die Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

Sie können mithilfe von Amazon Redshift-Metadaten, statistischen Informationen oder diesen beiden Optionen Sortier- und Verteilungsschlüssel auswählen. Wählen Sie auf der Registerkarte Optimierungsstrategien für die erste Schlüsselauswahlstrategie eine der folgenden Optionen aus:

- Metadaten verwenden, statistische Informationen ignorieren
- Metadaten ignorieren, statistische Informationen verwenden
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0—100) ein. Diese Werte definieren das Gewicht jeder Strategie. AWS SCT definiert anhand dieser Gewichtungswerte, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie Kleine Tabellen finden definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet

den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

Sie können nicht nur das Gewicht für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung.

- Geben Sie im Feld Sortierschlüsselspaltenlimit die maximale Anzahl von Spalten in den Sortierschlüssel ein.
- Geben Sie für schiefen Schwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCTschließt Spalten, deren Verzerrungswert größer als der Schwellenwert ist, aus der Liste der Kandidaten für den Verteilungsschlüssel aus. AWS SCTdefiniert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des häufigsten Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top-N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Auf der Registerkarte Optimierungsstrategien können Sie auch die Größe kleiner Tabellen für die Strategie Kleine Tabellen suchen definieren. Geben Sie für Minimale Tabellenzeilenanzahl und Maximale Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCTwendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Verwendung von Vertica als Quelle für AWS SCT

Sie können AWS SCT es verwenden, um Schemas, Codeobjekte und Anwendungscode von Vertica nach Amazon Redshift zu konvertieren.

Privilegien für Vertica als Quelle

Die folgenden Rechte sind erforderlich, um Vertica als Quelle verwenden zu können:

- USAGE ON SCHEMA *<schema_name>*
- USAGE ON SCHEMA PUBLIC
- SELECT ON ALL TABLES IN SCHEMA *<schema_name>*

- SELECT ON ALL SEQUENCES IN SCHEMA *<schema_name>*
- EXECUTE ON ALL FUNCTIONS IN SCHEMA *<schema_name>*
- EXECUTE ON PROCEDURE *<schema_name.procedure_name(procedure_signature)>*

Ersetzen Sie die Platzhalter im vorherigen Beispiel wie folgt:

- Ersetzen Sie *schema_name* durch den Namen des Quellschemas.
- Ersetzen Sie *procedure name durch den Namen* einer Quellprozedur. Wiederholen Sie den Zuschuss für jedes Verfahren, das Sie konvertieren.
- Ersetzen Sie *procedure_signature* durch die kommagetrennte Liste von Prozedurargumenttypen.

Verbindung zu Vertica als Quelle herstellen

Gehen Sie wie nachfolgend beschrieben vor, um mit dem AWS Schema Conversion Tool eine Verbindung zu Ihrer Vertica-Quelldatenbank herzustellen.

So stellen Sie eine Verbindung zu einer Vertica-Quelldatenbank her

1. Wählen Sie in der AWS Schema Conversion Tool Quelle hinzufügen aus.
2. Wählen Sie Vertica und dann Weiter.

Das Dialogfeld Quelle hinzufügen wird angezeigt.

3. Geben Sie unter Verbindungsname einen Namen für Ihre Datenbank ein. AWS SCT zeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. Verwenden Sie Datenbankmeldeinformationen von AWS Secrets Manager oder geben Sie sie manuell ein:
 - Gehen Sie wie folgt vor, um die Datenbankmeldeinformationen von Secrets Manager zu verwenden:
 1. Wählen Sie für AWSGeheim den Namen des Geheimnisses.
 2. Wählen Sie Auffüllen, um automatisch alle Werte im Datenbankverbindungsdialogfeld von Secrets Manager auszufüllen.

Hinweise zur Verwendung von Datenbankmeldeinformationen von Secrets Manager finden Sie unter [Verwenden von AWS Secrets Manager](#).

- Gehen Sie wie folgt vor, um die Verbindungsinformationen zur Vertica-Quelldatenbank manuell einzugeben:

Parameter	Action
Server name	Geben Sie den DNS-Namen (Domain Name System) oder die IP-Adresse Ihres Quelldatenbankservers ein.
Server port	Geben Sie den Port ein, der für die Verbindung zum Quelldatenbankserver verwendet wird.
Datenbank	Geben Sie den Namen der Vertica-Datenbank ein.
User name und Password	<p>Geben Sie die Datenbankanmeldeinformationen ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.</p> <p>AWS SCT verwendet das Passwort nur dann, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, wenn Sie sich in einem Projekt dafür entscheiden, eine Verbindung zu Ihrer Datenbank herzustellen. Um zu verhindern, dass das Passwort für Ihre Quelldatenbank preisgegeben wird, wird das Passwort standardmäßig AWS SCT nicht gespeichert. Wenn Sie Ihr AWS SCT-Projekt schließen und erneut öffnen, müssen Sie das Passwort für die Verbindung zur Quelldatenbank bei Bedarf erneut eingeben.</p>

Parameter	Action
Use SSL	<p>Wählen Sie diese Option, um Secure Sockets Layer (SSL) für die Verbindung mit Ihrer Datenbank zu verwenden . Geben Sie gegebenenfalls die folgenden zusätzlichen Informationen auf der Registerkarte SSL an:</p> <ul style="list-style-type: none">• Serverzertifikat überprüfen: Wählen Sie diese Option, um das Serverzertifikat mithilfe eines Trust Stores zu überprüfen.• Trust Store: Ein Trust Store, den Sie in den globalen Einstellungen einrichten.• Schlüsselspeicher: Ein Schlüsselspeicher, den Sie in den globalen Einstellungen einrichten.
Passwort speichern	<p>AWS SCT erstellt einen sicheren Bereich, um SSL-Zertifikate und Datenbankpasswörter zu speichern. Wenn Sie diese Option aktivieren, können Sie das Datenbankkennwort speichern und schnell eine Verbindung zur Datenbank herstellen, ohne das Passwort eingeben zu müssen.</p>
Vertikaler Treiberpfad	<p>Geben Sie den Pfad zum Treiber ein, der für die Verbindung zur Quelldatenbank verwendet wird. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen.</p> <p>Wenn Sie den Treiberpfad in den globalen Projekteinstellungen speichern, wird der Pfad im Verbindungsdialogfeld nicht angezeigt. Weitere Informationen finden Sie unter Speichern von Treiberpfaden in den globalen Einstellungen.</p>

5. Wählen Sie Verbindung testen, um zu überprüfen, ob eine Verbindung zu Ihrer Quelldatenbank hergestellt werden kann.
6. Wählen Sie Verbinden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen.

Einstellungen für die Konvertierung von Vertica nach Amazon Redshift

Um die Konvertierungseinstellungen von Vertica nach Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Vertica und dann Vertica — Amazon Redshift aus. AWS SCT zeigt alle verfügbaren Einstellungen für die Konvertierung von Vertica nach Amazon Redshift an.

Die Konvertierungseinstellungen von Vertica nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um die Anzahl der Kommentare mit Aktionselementen im konvertierten Code zu begrenzen.

Wählen Sie unter Kommentare im konvertierten Code für die Aktionspunkte mit dem ausgewählten Schweregrad und höher den Schweregrad der Aktionspunkte aus. AWS SCT fügt dem konvertierten Code Kommentare für Aktionspunkte mit dem ausgewählten Schweregrad und höher hinzu.

Um beispielsweise die Anzahl der Kommentare in Ihrem konvertierten Code zu minimieren, wählen Sie Nur Fehler aus. Um Kommentare zu allen Aktionselementen in Ihren konvertierten Code aufzunehmen, wählen Sie Alle Nachrichten.

- Um die maximale Anzahl von Tabellen festzulegen, die für Ihren Amazon Redshift-Zielcluster gelten AWS SCT können.

Wählen Sie unter Die maximale Anzahl von Tabellen für den Amazon Redshift-Cluster die Anzahl der Tabellen aus, die für Ihren Amazon Redshift-Cluster gelten AWS SCT können.

Amazon Redshift verfügt über Kontingente, die die Verwendung von Tabellen für verschiedene Cluster-Knotentypen einschränken. Wenn Sie Auto wählen, AWS SCT bestimmt es je nach Knotentyp die Anzahl der Tabellen, die auf Ihren Amazon Redshift-Ziel-Cluster angewendet werden sollen. Wählen Sie den Wert optional manuell aus. Weitere Informationen finden Sie unter [Kontingente und Limits in Amazon Redshift](#) im Verwaltungshandbuch zu Amazon Redshift.

AWS SCT konvertiert alle Ihre Quelltabellen, auch wenn dies mehr ist, als Ihr Amazon Redshift-Cluster speichern kann. AWS SCT speichert den konvertierten Code in Ihrem Projekt und wendet ihn nicht auf die Zieldatenbank an. Wenn Sie bei der Anwendung des konvertierten Codes das Amazon Redshift-Cluster-Kontingent für die Tabellen erreichen, wird eine Warnmeldung AWS SCT angezeigt. AWS SCT wendet Tabellen außerdem auf Ihren Amazon Redshift-Zielcluster an, bis die Anzahl der Tabellen das Limit erreicht.

- Um Partitionen der Quelltable in separate Tabellen in Amazon Redshift zu migrieren. Wählen Sie dazu die Ansicht UNION ALL verwenden aus und geben Sie die maximale Anzahl von Zieltabellen ein, die für eine einzelne Quelltable erstellt werden AWS SCT können.

Amazon Redshift unterstützt keine Tabellenpartitionierung. Um dieses Verhalten zu emulieren und Abfragen schneller ausführen zu lassen, AWS SCT können Sie jede Partition Ihrer Quelltable in eine separate Table in Amazon Redshift migrieren. Erstellen Sie AWS SCT dann eine Ansicht, die Daten aus all diesen Tabellen enthält.

AWS SCTbestimmt automatisch die Anzahl der Partitionen in Ihrer Quelltable. Je nach Art der Partitionierung der Quelltable kann diese Zahl das Kontingent für die Tabellen überschreiten, das Sie auf Ihren Amazon Redshift-Cluster anwenden können. Um zu vermeiden, dass dieses Kontingent erreicht wird, geben Sie die maximale Anzahl von Zieltabellen ein, die für Partitionen einer einzelnen Quelltable erstellt werden AWS SCT können. Die Standardoption ist 368 Tabellen, was einer Partition für 366 Tage eines Jahres und zwei Tabellen für NO RANGE und UNKNOWN Partitionen entspricht.

- Um die Komprimierung auf Amazon Redshift-Tabellenspalten anzuwenden. Wählen Sie dazu die Option Kompressionskodierung verwenden aus.

AWS SCTweist Spalten mithilfe des standardmäßigen Amazon Redshift-Algorithmus automatisch eine Kompressionskodierung zu. Weitere Informationen finden Sie unter [Kompressionskodierungen](#) im Amazon Redshift Database Developer Guide.

Standardmäßig wendet Amazon Redshift keine Komprimierung auf Spalten an, die als Sortier- und Verteilungsschlüssel definiert sind. Sie können dieses Verhalten ändern und die Komprimierung auf diese Spalten anwenden. Wählen Sie dazu die Option Kompressionskodierung für KEY-Spalten verwenden aus. Sie können diese Option nur auswählen, wenn Sie die Option Kompressionskodierung verwenden ausgewählt haben.

Einstellungen für die Konversionsoptimierung von Vertica nach Amazon Redshift

Um die Einstellungen für die Konvertierungsoptimierung von Vertica nach Amazon Redshift zu bearbeiten, wählen Sie Einstellungen in AWS SCT und dann Konvertierungseinstellungen. Wählen Sie in der oberen Liste Vertica und dann Vertica — Amazon Redshift aus. Wählen Sie im linken Bereich Optimierungsstrategien aus. AWS SCTzeigt die Einstellungen zur Konvertierungsoptimierung für die Konvertierung von Vertica nach Amazon Redshift an.

Die Einstellungen für die Konversionsoptimierung von Vertica nach Amazon Redshift AWS SCT beinhalten Optionen für Folgendes:

- Um mit der automatischen Tabellenoptimierung zu arbeiten. Wählen Sie dazu die automatische Tabellenoptimierung von Amazon Redshift verwenden aus.

Die automatische Tabellenoptimierung ist ein Selbstoptimierungsprozess in Amazon Redshift, der das Design von Tabellen automatisch optimiert. Weitere Informationen finden Sie unter [Arbeiten mit automatischer Tabellenoptimierung](#) im Amazon Redshift Database Developer Guide.

Wenn Sie sich nur auf die automatische Tabellenoptimierung verlassen möchten, wählen Sie „Keine“ als Strategie für die anfängliche Schlüsselauswahl aus.

- Um die Sortier- und Verteilungsschlüssel anhand Ihrer Strategie auszuwählen.

Sie können mithilfe von Amazon Redshift-Metadaten, statistischen Informationen oder diesen beiden Optionen Sortier- und Verteilungsschlüssel auswählen. Wählen Sie auf der Registerkarte Optimierungsstrategien für die erste Schlüsselauswahlstrategie eine der folgenden Optionen aus:

- Metadaten verwenden, statistische Informationen ignorieren
- Metadaten ignorieren, statistische Informationen verwenden
- Verwenden Sie Metadaten und statistische Informationen

Je nachdem, welche Option Sie wählen, können Sie Optimierungsstrategien auswählen. Geben Sie dann für jede Strategie den Wert (0–100) ein. Diese Werte definieren das Gewicht jeder Strategie. AWS SCT definiert anhand dieser Gewichtungswerte, wie sich jede Regel auf die Wahl der Verteilungs- und Sortierschlüssel auswirkt. Die Standardwerte basieren auf den bewährten Methoden für die AWS Migration.

Sie können die Größe kleiner Tabellen für die Strategie Kleine Tabellen finden definieren. Geben Sie für Min. Tabellenzeilenanzahl und Max. Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu definieren. AWS SCT wendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

- Um Strategiedetails zu konfigurieren.

Sie können nicht nur das Gewicht für jede Optimierungsstrategie definieren, sondern auch die Optimierungseinstellungen konfigurieren. Wählen Sie dazu Konversionsoptimierung.

- Geben Sie im Feld Sortierschlüsselspaltenlimit die maximale Anzahl von Spalten in den Sortierschlüssel ein.

- Geben Sie für schiefen Schwellenwert den Prozentsatz (0—100) eines schiefen Werts für eine Spalte ein. AWS SCTschließt Spalten, deren Verzerrungswert größer als der Schwellenwert ist, aus der Liste der Kandidaten für den Verteilungsschlüssel aus. AWS SCTdefiniert den schiefen Wert für eine Spalte als das prozentuale Verhältnis der Anzahl der Vorkommen des häufigsten Werts zur Gesamtzahl der Datensätze.
- Geben Sie für die Top-N-Abfragen aus der Tabelle mit dem Abfrageverlauf die Anzahl (1—100) der am häufigsten zu analysierenden Abfragen ein.
- Wählen Sie unter Statistikbenutzer auswählen den Datenbankbenutzer aus, für den Sie die Abfragestatistiken analysieren möchten.

Auf der Registerkarte Optimierungsstrategien können Sie auch die Größe kleiner Tabellen für die Strategie Kleine Tabellen suchen definieren. Geben Sie für Minimale Tabellenzeilenanzahl und Maximale Tabellenzeilenanzahl die minimale und maximale Anzahl von Zeilen in einer Tabelle ein, um sie als kleine Tabelle zu betrachten. AWS SCTwendet den ALL Verteilungsstil auf kleine Tabellen an. In diesem Fall wird eine Kopie der gesamten Tabelle an jeden Knoten verteilt.

Erstellen von Zuordnungsregeln in AWS SCT

Sie können mehrere Quell- und Zieldatenbanken in einem einzigen AWS SCT Projekt hinzufügen. Dies vereinfacht die Verwaltung von Projekten, wenn Sie mehrere Datenbanken auf verschiedene Zielplattformen migrieren.

Nachdem Sie ein neues Projekt erstellt und Quell- und Zieldatenbanken hinzugefügt haben, erstellen Sie Zuordnungsregeln. AWS SCT erfordert mindestens eine Zuordnungsregel, um einen Migrationsbewertungsbericht zu erstellen und Datenbankschemas zu konvertieren.

Eine Zuordnungsregel beschreibt ein Quell-Ziel-Paar, das ein Quelldatenbankschema oder eine Quelldatenbank und eine Zieldatenbankplattform umfasst. Sie können mehrere Zuordnungsregeln in einem einzigen AWS SCT Projekt erstellen. Verwenden Sie Zuordnungsregeln, um jedes Quelldatenbankschema auf die richtige Zieldatenbankplattform zu konvertieren.

Um den Namen Ihres Schemas im konvertierten Code zu ändern, richten Sie eine Migrationsregel ein. Mit Migrationsregeln können Sie beispielsweise Ihr Schema umbenennen, Objektnamen ein Präfix hinzufügen, die Spaltensortierung ändern oder Datentypen ändern. Um diese Änderungen auf Ihren konvertierten Code anzuwenden, stellen Sie sicher, dass Sie Migrationsregeln erstellen, bevor Sie Ihr Quellschema konvertieren. Weitere Informationen finden Sie unter [Migrationsregeln erstellen](#).

Sie können Zuordnungsregeln nur für unterstützte Datenbankkonvertierungspaare erstellen. Eine Liste der unterstützten Konvertierungspaare finden Sie unter [Quellen für AWS SCT](#).

Wenn Sie ein Projekt öffnen, das in AWS SCT Version 1.0.655 oder früher gespeichert wurde, AWS SCT werden automatisch Regeln für die Zuordnung aller Quelldatenbankschemas zur Zieldatenbankplattform erstellt. Um weitere Zieldatenbankplattformen hinzuzufügen, löschen Sie vorhandene Zuordnungsregeln und erstellen Sie dann neue Zuordnungsregeln.

Themen

- [Neue Zuordnungsregel hinzufügen](#)
- [Verwaltung von Zuordnungsregeln](#)
- [Verwendung virtueller Ziele](#)
- [Einschränkungen bei der Verwendung mehrerer Server in einem einzigen AWS SCT Projekt](#)

Neue Zuordnungsregel hinzufügen

Sie können mehrere Zuordnungsregeln in einem einzigen Projekt erstellen. AWS SCT speichert Zuordnungsregeln als Teil Ihres Projekts. Gehen Sie bei geöffnetem Projekt wie folgt vor, um eine neue Zuordnungsregel hinzuzufügen.

So erstellen Sie Zuweisungsregeln

1. Wählen Sie im Menü Ansicht die Option Mapping-Ansicht aus.
2. Wählen Sie im linken Bereich ein Schema oder eine Datenbank aus, die Sie der Zuordnungsregel hinzufügen möchten.
3. Wählen Sie im rechten Bereich eine Zieldatenbankplattform für das ausgewählte Quellschema oder die ausgewählte Datenbank aus.

Sie können eine virtuelle Datenbankplattform als Ziel wählen. Weitere Informationen finden Sie unter [Verwendung virtueller Ziele](#).

4. Wählen Sie Mapping erstellen aus.

AWS SCT fügt diese neue Zuordnungsregel zur Liste der Serverzuordnungen hinzu.

Fügen Sie Zuordnungsregeln für alle Konvertierungspaare hinzu. Um einen Bewertungsbericht zu erstellen oder Datenbankschemas zu konvertieren, wählen Sie im Menü Ansicht die Option Hauptansicht.

AWS SCT hebt alle Schemaobjekte, die Teil einer Zuordnungsregel sind, fett hervor.

Verwaltung von Zuordnungsregeln

Sie können bestehende Zuordnungsregeln filtern oder löschen und Ihrem AWS Schema Conversion Tool (AWS SCT) -Projekt eine neue Zuordnungsregel hinzufügen.

Wenn Sie eine Zuordnungsregel für die gesamte Quelldatenbank erstellen, AWS SCT wird eine Zuordnungsregel für jedes Quelldatenbankschema erstellt. Bei Projekten, die Dutzende von Schemas oder sogar Datenbanken beinhalten, kann es schwierig sein zu verstehen, welches Ziel für ein bestimmtes Schema verwendet wird. Um schnell eine Zuordnungsregel für Ihr Schema zu finden, verwenden Sie eine oder mehrere der folgenden Filteroptionen in AWS SCT.

Um Zuordnungsregeln zu filtern

1. Wählen Sie im Menü Ansicht die Option Mapping-Ansicht aus.
2. Wählen Sie für Quellserver die Quelldatenbank aus.

Die Standardeinstellung des Filters ist Alle, was bedeutet, dass Zuordnungsregeln für alle Quelldatenbanken AWS SCT angezeigt werden.

3. Geben Sie für Quellschema den Namen des Quellschemas ein. Verwenden Sie den Prozentsatz (%) als Platzhalter, um eine beliebige Anzahl von Symbolen im Schemanamen zu ersetzen.

Der Standardfilter ist der Platzhalter%, was bedeutet, dass Zuordnungsregeln für alle Schemanamen der Quelldatenbank AWS SCT angezeigt werden.

4. Wählen Sie unter Hat Migrationsregeln die Option Ja aus, um die Zuordnungsregeln anzuzeigen, für die die Datenmigrationsregeln erstellt wurden. Wählen Sie Nein, um Zuordnungsregeln anzuzeigen, für die keine Datenmigrationsregeln gelten. Weitere Informationen finden Sie unter [Regeln für die Datenmigration erstellen in AWS SCT](#).

Die Standardeinstellung des Filters ist Alle, was bedeutet, dass alle Zuordnungsregeln AWS SCT angezeigt werden.

5. Wählen Sie für Zielsever die Zieldatenbank aus.

Die Standardeinstellung des Filters ist Alle, was bedeutet, dass Zuordnungsregeln für alle Zieldatenbanken AWS SCT angezeigt werden.

Gehen Sie bei geöffnetem Projekt wie folgt vor, um eine Zuordnungsregel zu löschen. Weitere Informationen zum Hinzufügen von Zuordnungsregeln finden Sie unter [Neue Zuordnungsregel hinzufügen](#).

Um Zuordnungsregeln zu löschen

1. Wählen Sie im Menü Ansicht die Option Mapping-Ansicht aus.
2. Wählen Sie für Serverzuordnungen die zu löschenden Zuordnungsregeln aus.
3. Wählen Sie Ausgewählte Mappings löschen.

AWS SCT löscht die ausgewählten Zuordnungsregeln.

Verwendung virtueller Ziele

Sie können sehen, wie Ihr Quelldatenbankschema in jede unterstützte Zieldatenbankplattform AWS SCT konvertiert wird. Dazu müssen Sie keine Verbindung zu einer vorhandenen Zieldatenbank herstellen. Stattdessen können Sie im rechten Bereich eine virtuelle Zieldatenbankplattform auswählen, wenn Sie eine Zuordnungsregel erstellen. Weitere Informationen finden Sie unter [Neue Zuordnungsregel hinzufügen](#). Stellen Sie sicher, dass Sie die Server, NoSQL-Cluster und ETL-Knoten im rechten Bereich erweitern, um die Liste der virtuellen Zieldatenbankplattformen zu sehen.

AWS SCT unterstützt die folgenden virtuellen Zieldatenbankplattformen:

- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon DynamoDB
- Amazon Redshift
- Amazon Redshift und AWS Glue
- AWS Glue
- AWS Glue Studio
- Babelfish für Aurora PostgreSQL
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Wenn Sie Babelfish for Aurora PostgreSQL als Zieldatenbankplattform verwenden, können Sie nur einen Bewertungsbericht zur Datenbankmigration erstellen. Weitere Informationen finden Sie unter [the section called “Berichte über die Migrationsbewertung”](#).

Wenn Sie eine virtuelle Zieldatenbankplattform verwenden, können Sie konvertierten Code in einer Datei speichern. Weitere Informationen finden Sie unter [the section called “Speichern Ihres konvertierten Schemas”](#).

Einschränkungen bei der Verwendung mehrerer Server in einem einzigen AWS SCT Projekt

Die folgenden Einschränkungen gelten bei der Konvertierung von Schemas mithilfe mehrerer Server in einem einzigen AWS SCT Projekt:

- Sie können denselben Server nur einmal zu einem Projekt hinzufügen.
- Sie können Serverschemas nicht einem bestimmten Zielschema zuordnen, sondern nur einem Zielserver. AWS SCT erstellt das Zielschema während der Konvertierung.
- Sie können dem Zielserver keine Quellobjekte auf niedrigerer Ebene zuordnen.
- Sie können ein Quellschema nur einem Zielserver in einem Projekt zuordnen.
- Stellen Sie sicher, dass Sie eine Quelle einem Zielserver zuordnen, um einen Bewertungsbericht zu erstellen, Schemas zu konvertieren oder Daten zu extrahieren.

Erstellen von Konvertierungsberichten

Wenn Sie die Konvertierung einer Datenbank planen, ist es hilfreich, einige Berichte erstellen. So können Sie besser überblicken, welche Aktivitäten erforderlich sind. Sie können Berichte mit AWS Schema Conversion Tool erstellen.

Sie können AWS SCT verwenden, um einen Bewertungsbericht zur Datenbankmigration zu erstellen. Dieser Bericht fasst alle Aufgaben der Schemakonvertierung zusammen und führt detailliert die Elemente auf, die nicht automatisch in die Zieldatenbank konvertiert werden können. Sie können diesen Bericht verwenden, um zu bewerten, welche Werte des Projekts mithilfe von AWS SCT und was Sie sonst noch benötigen, um die Konvertierung abzuschließen. Verwenden Sie zum Erstellen eines Bewertungsberichts die Option Create Report (Bericht erstellen) aus dem Kontextmenü (rechte Maustaste) der Datenbank in AWS SCT.

Themen

- [Erstellung von Berichten zur Migrationsbewertung mit AWS SCT](#)

Erstellung von Berichten zur Migrationsbewertung mit AWS SCT

Ein wichtiger Teil des AWS Schema Conversion Tool ist der Bewertungsbericht, den es erstellt, um die Komplexität Ihrer Schemakonvertierung abzuschätzen. Dieser Bewertungsbericht zur Datenbankmigration fasst alle Aufgaben zur Schemakonvertierung zusammen und beschreibt detailliert die Aktionspunkte für ein Schema, das nicht in die DB-Engine Ihrer Ziel-DB-Instance konvertiert werden kann. Sie können den Bericht in der Anwendung anzeigen oder als CSV-Datei oder PDF-Datei exportieren.

Wenn Sie mehrere Quell- und Zieldatenbanken in einem einzigen Projekt hinzufügen, werden die AWS SCT Berichte für alle Konvertierungspaare in einem Bewertungsbericht für die Datenbankmigration zusammengefasst.

Sie können virtuelle Zieldatenbankplattformen verwenden, um einen Bewertungsbericht zu erstellen und die Komplexität der Migration zu einer ausgewählten Datenbankplattform zu verstehen. In diesem Fall müssen Sie keine Verbindung mit Ihrer Zieldatenbankplattform herstellen. Beispielsweise können Sie Babelfish für Aurora PostgreSQL als virtuelle Zieldatenbankplattform verwenden, um einen Bewertungsbericht zur Datenbankmigration zu erstellen. Weitere Informationen zu virtuellen Zieldatenbankplattformen finden Sie unter [the section called "Virtuelle Ziele"](#).

Der Bericht zur Bewertung der Datenbankmigration enthält die folgenden Elemente:

- Zusammenfassung
- Lizenzauswertung
- Cloud-Support, mit Angabe aller Funktionen in der Quelldatenbank, die auf dem Ziel nicht verfügbar sind.
- Empfehlungen, einschließlich Konvertierung von Server-Objekten, Backup-Vorschläge und verknüpfte Server-Änderungen

Außerdem enthält der Bericht eine Aufwandsschätzung, da für die nicht automatisch konvertierbaren Elemente entsprechender Code für die DB-Ziel-Instance geschrieben werden muss.

Wenn Sie Ihr vorhandenes SchemaAWS SCT zu einer Amazon RDS-DB-Instance migrieren, können Sie den Bericht verwenden, um die Anforderungen für den Umzug in dieAWS Cloud zu analysieren und Ihren Lizenztyp zu ändern.

Themen

- [Erstellen eines Berichts über die Bewertung der Datenbankmigration](#)
- [Den Bewertungsbericht anzeigen](#)
- [Speichern des Bewertungsberichts](#)
- [Konfiguration des Bewertungsberichts](#)
- [Erstellung eines Multiserver-Bewertungsberichts für die Datenbankmigration](#)

Erstellen eines Berichts über die Bewertung der Datenbankmigration

Führen Sie die folgenden Schritte aus, um einen Bewertungsbericht zur Datenbankmigration zu erstellen.

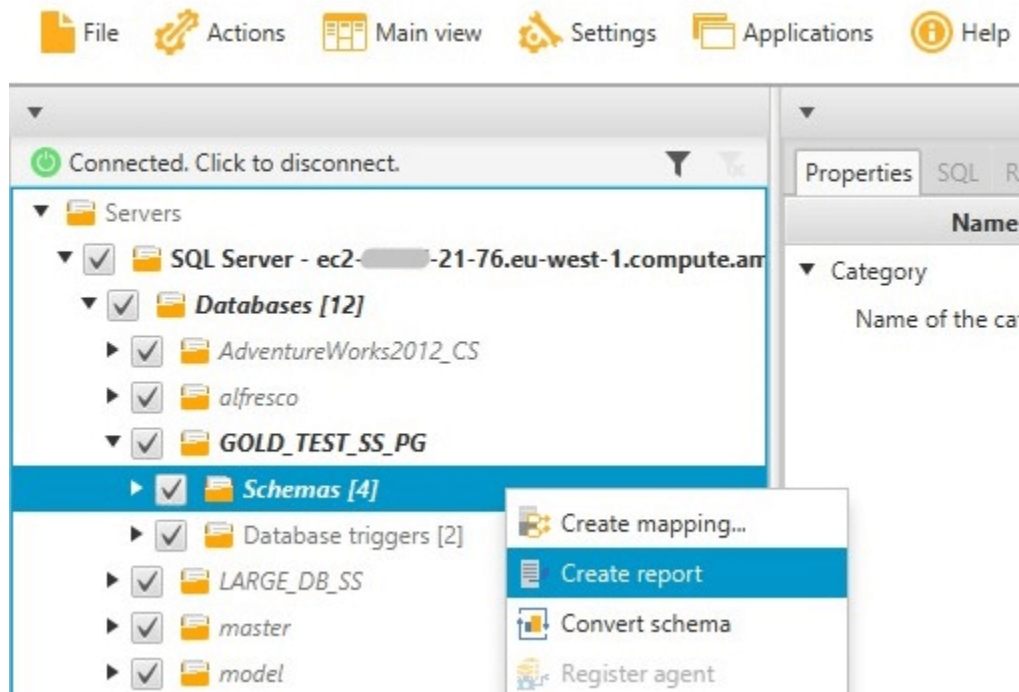
So erstellen Sie einen Bewertungsbericht zur Datenbankmigration

1. Stellen Sie sicher, dass Sie eine Zuordnungsregel für das Quelldatenbankschema erstellt haben, für das Sie einen Bewertungsbericht erstellen möchten. Weitere Informationen finden Sie unter [Neue Zuordnungsregel hinzufügen](#).
2. Wählen Sie im Menü Ansicht die Option Hauptansicht.
3. Wählen Sie im linken Bereich, in dem Ihr Quelldatenbankschema angezeigt wird, ein Schemaobjekt aus, für das Sie einen Bewertungsbericht erstellen möchten. Um mehrere

Datenbankschemas in den Bericht aufzunehmen, wählen Sie den übergeordneten Knoten aus, zum Beispiel Schemas.

Stellen Sie sicher, dass Sie die Kontrollkästchen für alle Schemaobjekte aktiviert haben, für die Sie einen Bewertungsbericht erstellen möchten.

- Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie dann Bericht erstellen aus.



Den Bewertungsbericht anzeigen

Nach der Erstellung des Bewertungsberichts wird die entsprechende Ansicht mit folgenden Registerkarten geöffnet:

- Übersicht
- Action Items (Aktionselemente)

Die Registerkarte Summary (Zusammenfassung) enthält Elemente, die automatisch konvertiert oder nicht konvertiert wurden.

Die Registerkarte Action Items (Aktionselemente) zeigt Elemente an, die nicht automatisch konvertiert werden konnten, sowie Empfehlungen zur Handhabung dieser Elemente.

Themen

- [Zusammenfassung des Bewertungsberichts](#)
- [Maßnahmenpunkte des Bewertungsberichts](#)
- [Warnmeldung des Bewertungsberichts](#)

Zusammenfassung des Bewertungsberichts

Auf der Registerkarte Summary wird eine Zusammenfassung des Berichts zur Bewertung der Datenbankmigration angezeigt. Es werden sowohl die automatisch konvertierten Elemente als auch Elemente, die nicht automatisch konvertiert wurden, angezeigt.

Summary | Action items
Save to CSV | Save to PDF

Database migration assessment report

Source database: GOLD_TEST_SS_PG: >21-76.eu-west-1.compute.amazonaws.com/GOLD_TEST_SS_PG:1433
 Microsoft SQL Server 2019 (RTM-CU10) (KB5001090) - 15.0.4123.1 (X64) Mar 22 2021 18:10:24
 Copyright (C) 2019 Microsoft Corporation
 Enterprise Edition: Core-based Licensing (64-bit) on Windows Server 2019 Datacenter 10.0 <X64> (Build 17763:) (Hypervisor)
 Case sensitivity: OFF

Executive summary

We completed the analysis of your Microsoft SQL Server source database and estimate that 90% of the database storage objects and 77% of database code objects can be converted automatically or with minimal changes if you select Amazon RDS for PostgreSQL as your migration target. Database storage objects include schemas, tables, table constraints, indexes, types, table types, sequences, synonyms and xml schema collections. Database code objects include triggers, views, procedures, scalar functions, inline functions, table-valued functions and database triggers. Based on the source code syntax analysis, we estimate 94% (based on # lines of code) of your code can be converted to Amazon RDS for PostgreSQL automatically. To complete the migration, we recommend 3,300 conversion action(s) ranging from simple tasks to medium-complexity actions to complex conversion actions.

Migration guidance for database objects that could not be converted automatically can be found [here](#)

Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

Figure: Conversion statistics for database storage objects

Object Type	Count	Automatically converted	Simple actions	Medium-complexity actions	Complex actions
Schema (4: 4/0/0/0)	4	100%	0%	0%	0%
Table (323: 276/8/2/37)	323	85%	2%	11%	0%
Constraint (157: 152/2/0/3)	157	97%	2%	0%	0%
Index (63: 36/22/0/5)	63	57%	35%	8%	0%
Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Sequence (14: 7/7/0/0)	14	50%	50%	0%	0%
Synonym (5: 0/0/0/5)	5	0%	0%	100%	0%
Table Type (7: 7/0/0/0)	7	100%	0%	0%	0%
Xml schema collection (5: 1/0/0/4)	5	20%	80%	0%	0%

Für die Schemaelemente, die nicht automatisch in die Zieldatenbank-Engine konvertiert werden können, enthält die Zusammenfassung eine Aufwandsschätzung zur Erstellung von Schemaelementen in Ihrer Ziel-DB-Instance, die mit denen in der Quelle gleichgesetzt werden können.

Im Bericht wird der geschätzte Zeitaufwand für die Konvertierung dieser Schemaelemente wie folgt kategorisiert:

- **Einfach** — Aktionen, die in weniger als zwei Stunden abgeschlossen werden können.
- **Mittel** — Aktionen, die komplexer sind und in zwei bis sechs Stunden abgeschlossen werden können.
- **Signifikant** — Aktionen, die sehr komplex sind und mehr als sechs Stunden in Anspruch nehmen.

Der Abschnitt Lizenzbewertung und Cloud-Support enthält Informationen zum Verschieben Ihres vorhandenen lokalen Datenbankschemas in eine Amazon RDS-DB-Instance, auf der dieselbe Engine ausgeführt wird. Wenn Sie beispielsweise Lizenztypen ändern möchten, finden Sie in diesem Abschnitt Informationen dazu, welche Funktionen Ihrer aktuellen Datenbank entfernt werden sollten.

License evaluation

Our analysis shows that current schema uses the following Enterprise Edition features unavailable in Standard Edition.

Feature	Description
Database In-Memory	Oracle Database In-Memory optimizes both analytics and mixed workload OLTP, delivering outstanding performance for transactions while simultaneously supporting real-time analytics, business intelligence, and reports.
Materialized View Query Rewrite	Oracle Database employs an extremely powerful process called query rewrite to quickly answer the query using materialized views.
Partitioning	Partitioning is powerful functionality that allows tables, indexes, and index-organized tables to be subdivided into smaller pieces, enabling these database objects to be managed and accessed at a finer level of granularity.
Oracle Advanced Security/TDE	Oracle Advanced Security provides two important preventive controls to protect sensitive data at the source: encryption and redaction. Together, these two controls form the foundation of Oracle's defense-in-depth, multi-layered database security solution.

If you choose Standard Edition as your migration target, remove dependencies on these features.

Cloud support

Our analysis shows that your current schema uses the following features that require configuration steps in Amazon RDS for Oracle.

Feature	Description
Locator	Oracle Locator provides capabilities that are typically required to support internet and wireless service-based applications and partner-based GIS solutions. Oracle Locator is a limited subset of Oracle Spatial. Please read prerequisites and configuration steps in the next article: Oracle Locator .
Spatial	Oracle Spatial provides a SQL schema and functions that facilitate the storage, retrieval, update, and query of collections of spatial data in an Oracle database. Please read prerequisites and configuration steps in the next article: Oracle Spatial .
Oracle XML DB	Oracle XML DB provides full support for all of the key XML standards, including XML Namespaces, DOM, XQuery, SQL/XML and XSLT. Amazon RDS for Oracle supports XML DB feature without the XML DB Protocol Server. Please read prerequisites and configuration steps in the next article: Oracle XML DB option .

If choose Amazon RDS for Oracle as your migration target, please follow the abovementioned steps to continue to use these features on the target database after migration completes.

Maßnahmenpunkte des Bewertungsberichts

Im Bewertungsbericht ist auch eine Registerkarte Action Items vorhanden. Diese Registerkarte enthält eine Liste von Elementen, die nicht automatisch in die Datenbank-Engine Ihrer Amazon RDS-DB-Zielinstanz konvertiert werden können. Wenn Sie ein Aktionselement aus der Liste auswählen, markiert AWS SCT das entsprechende Element im Schema, für das diese Aktion gilt.

Der Bericht enthält zudem Empfehlungen für die manuelle Konvertierung des Schemaelements. Beispielsweise zeigen Ihnen detaillierte Berichte für die Datenbank/das Schema nach Ausführung der Bewertung den Aufwand, der erforderlich ist, um die Empfehlungen für die Konvertierung von Aktionselementen zu entwerfen und umzusetzen. Weitere Informationen zum Festlegen der Handhabung von manuellen Konvertierungen finden Sie unter [Bearbeitung manueller Konvertierungen in AWS SCT](#).

The screenshot displays the AWS Schema Conversion Tool (SCT) interface. The top menu includes File, Actions, Assessment Report view, Settings, Applications, Help, Add source, and Add target. The main window is divided into several sections:

- Summary:** Shows the current connection status (Connected) and a list of servers.
- Issue List:** A list of issues identified during the conversion process, including:
 - Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required.
 - Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically.
 - Issue 794:** MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype.
 - Issue 826:** Check the default value for a DateTime variable.
 - Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision.
 - Issue 9997:** Unable to resolve objects.
 - Issue 690:** MySQL doesn't support table types.
 - Issue 811:** Unable to convert functions.
- Issue Details:** A detailed view of a selected issue (Issue 690) showing the recommended action: "Perform a manual conversion." Below this, the source SQL code is displayed:


```

1 create procedure POSITION_UPDATE_CASH_CGT_BULK
2   @InputPosNo tinyint readonly
3   , @posFlags bigint = 0
4   , @posFlagsMask bigint = 0
5 AS
6 update p
7 set   p.Flags = p.Flags & (~ @posFlagsMask ) | @posFlags
8 from Position p
9       inner join @InputPosNo ipn on p.PosNo = ipn.F_POSNO
10
11 return 0
      
```
- Target Amazon RDS for MySQL category: Schemas:** A table showing the target schema details:

Name	Value
Category	Schemas
Name of the category	Schemas

Warnmeldung des Bewertungsberichts

Um die Komplexität der Konvertierung zu einer anderen Datenbank-Engine beurteilen zu können, AWS SCT ist Zugriff auf Objekte in Ihrer Quelldatenbank erforderlich. Wenn SCT keine Bewertung durchführen kann, weil beim Scannen Probleme aufgetreten sind, wird eine Warnmeldung ausgegeben, die darauf hinweist, dass der Gesamtkonvertierungsprozentsatz reduziert wurde.

Warning!

We found that your source database may be configured not in correct way or you have not enough privileges for reading all necessary metadata. Please check your configuration and run report again. For more details please review [help documentation](#).

List of Action Items to review:

1. **Issue 9997** Unable to resolve objects (number of occurrences: 3)

Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually.

Im Folgenden sind die Gründe aufgeführt, aus denen beim Scannen Probleme AWS SCT auftreten können:

- Das mit der Datenbank verbundene Benutzerkonto hat nicht Zugriff auf alle benötigten Objekte.
- Ein im Schema zitiertes Objekt ist in der Datenbank nicht mehr vorhanden.
- SCT versucht, ein verschlüsseltes Objekt zu bewerten.

Weitere Informationen zu den für SCT erforderlichen Sicherheitsberechtigungen und -berechtigungen für Ihre Datenbank finden Sie im Abschnitt [Quellen für AWS SCT](#) zur entsprechenden Quelldatenbank in diesem Handbuch.

Speichern des Bewertungsberichts

Nachdem Sie [einen Bewertungsbericht zur Datenbankmigration erstellt](#) haben, können Sie eine lokale Kopie des Bewertungsberichts zur Datenbankmigration entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV) speichern.

So speichern Sie einen Bewertungsbericht zur Datenbankmigration als PDF-Datei

1. Wählen Sie im oberen Menü „Ansicht“ und dann „Ansicht des Bewertungsberichts“.
2. Wählen Sie die Registerkarte Summary aus.
3. Wählen Sie oben rechts Als PDF speichern.

So speichern Sie einen Bewertungsbericht zur Datenbankmigration als CSV-Datei

1. Wählen Sie im oberen Menü „Ansicht“ und dann „Ansicht des Bewertungsberichts“.
2. Wählen Sie die Registerkarte Summary aus.
3. Wählen Sie oben rechts In CSV speichern.

Die PDF-Datei enthält die Zusammenfassung und Informationen zu Aktionselementen, wie im folgenden Beispiel dargestellt.

Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

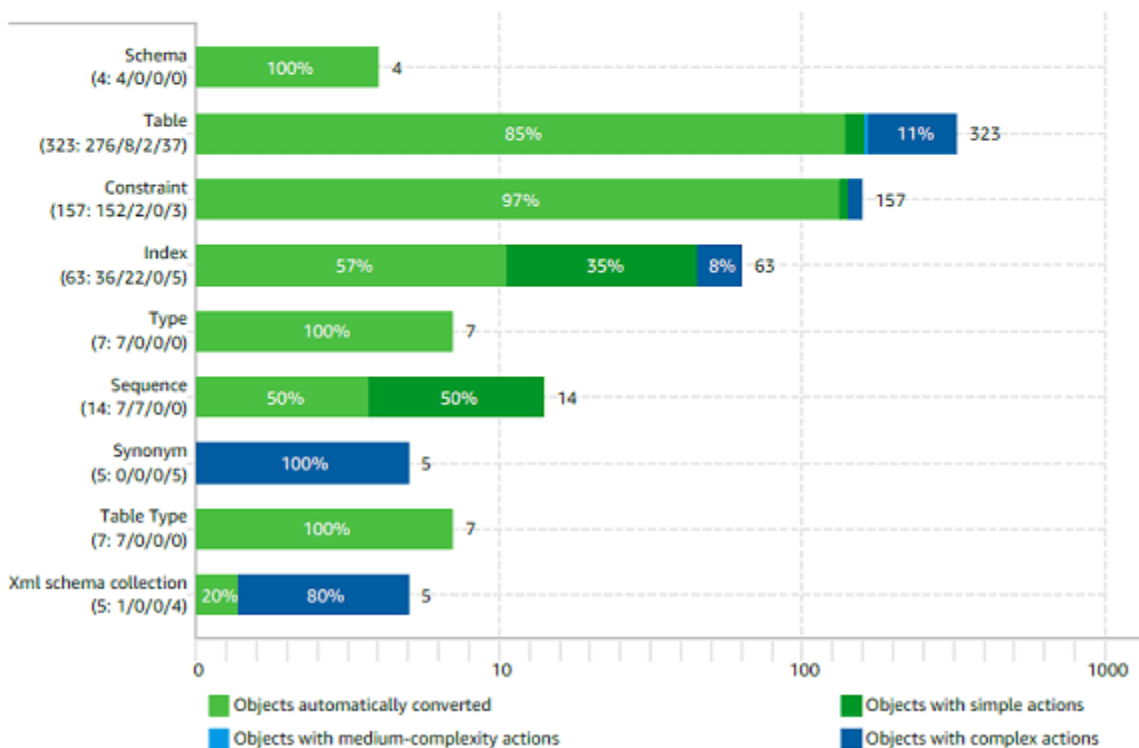
We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

Figure: Conversion statistics for database storage objects



Wenn Sie die Option In CSV speichern wählen, werden drei CSV-Dateien AWS SCT erstellt.

Die erste CSV-Datei enthält die folgenden Informationen zu den Aktionselementen:

- Kategorie
- Vorkommen — Der Dateiname, die Zeilennummer und die Position für das Element
- Nummer des Aktionspunkts

- Betreff
- Gruppe
- Beschreibung
- Referenzen zur Dokumentation
- Handlungsempfehlung
- Geschätzte Komplexität

Die zweite CSV-Datei enthält das `Action_Items_Summary` Suffix in ihrem Namen und enthält die Informationen über die Anzahl der Vorkommen aller Aktionselemente.

Im folgenden Beispiel geben die Werte in der Spalte Lernkurve den Aufwand an, der erforderlich ist, um einen Ansatz für die Konvertierung der einzelnen Aktionspunkte zu entwickeln. Die Werte in der Spalte Aufwand zur Konvertierung eines Vorkommens des Aktionspunkts geben den Aufwand an, der erforderlich ist, um die einzelnen Aktionspunkte zu konvertieren, wobei der entworfene Ansatz eingehalten wird. Die zur Angabe des erforderlichen Aufwandes verwendeten Werte basieren auf einer gewichteten Skala, die von niedrig (am geringsten) bis hoch (am meisten) reicht.

Schema	Action item	Number of occurrences	Learning curve efforts	Efforts to convert an occurrence of the action item
TEST.dbo	609	1	8	0.3
TEST.dbo	681	2	0.1	0.1
TEST.dbo	690	1	40	40
TEST.dbo	794	1	0	0.01
TEST.dbo	811	12	40	8
TEST.dbo	826	1	0	0.1
TEST.dbo	844	8	8	0.5
TEST.dbo	9997	3	0	0.3

Die dritte CSV-Datei enthält `Summary` ihren Namen und enthält die folgende Zusammenfassung:

- Kategorie
- Anzahl der Objekte
- Objekte automatisch konvertiert
- Objekte mit einfachen Aktionen
- Objekte mit Aktionen mittlerer Komplexität
- Objekte mit komplexen Aktionen
- Gesamtzahl der Codezeilen

Konfiguration des Bewertungsberichts

Sie können die Anzahl der Details konfigurieren, die in den Bewertungsberichten AWS SCT enthalten sind.

So konfigurieren Sie einen Bewertungsbericht zur Datenbankmigration

1. Wählen Sie im Menü Einstellungen die Option Globale Einstellungen und dann Bewertungsbericht aus.
2. Wählen Sie für das Vorkommen von Aktionspunkten die Option Nur die ersten fünf Probleme aus, um die Anzahl der Aktionspunkte eines einzelnen Typs im Bewertungsbericht zu begrenzen. Wählen Sie Alle Probleme, um alle Aktionspunkte jedes Typs in den Bewertungsbericht aufzunehmen.
3. Wählen Sie für analysierte SQL-Skriptdateien die Option Nicht mehr als **X** Dateien auflisten aus, um die Anzahl der SQL-Skriptdateien im Bewertungsbericht zu beschränken **X**. Geben Sie die Anzahl der Dateien ein. Wählen Sie Alle analysierten Dateien auflisten, um alle SQL-Skriptdateien in den Bewertungsbericht aufzunehmen.
4. Wählen Sie Berichte nach dem Speichern öffnen aus, um die Datei automatisch zu öffnen, nachdem Sie eine lokale Kopie des Bewertungsberichts zur Datenbankmigration gespeichert haben. Weitere Informationen finden Sie unter

[Nachdem Sie einen Bewertungsbericht zur Datenbankmigration erstellt haben, können Sie eine lokale Kopie des Bewertungsberichts zur Datenbankmigration entweder als PDF-Datei oder als Datei mit kommagetrennten Werten \(CSV\) speichern.](#)

[So speichern Sie einen Bewertungsbericht zur Datenbankmigration als PDF-Datei](#)

1. Wählen Sie im oberen Menü „Ansicht“ und dann „Ansicht des Bewertungsberichts“.
 2. Wählen Sie die Registerkarte Summary aus.
 3. Wählen Sie oben rechts Als PDF speichern.
-

[So speichern Sie einen Bewertungsbericht zur Datenbankmigration als CSV-Datei](#)

1. Wählen Sie im oberen Menü „Ansicht“ und dann „Ansicht des Bewertungsberichts“.
 2. Wählen Sie die Registerkarte Summary aus.
 3. Wählen Sie oben rechts In CSV speichern.
-

Die PDF-Datei enthält die Zusammenfassung und Informationen zu Aktionselementen, wie im folgenden Beispiel dargestellt.

Database objects with conversion actions for Amazon RDS for PostgreSQL

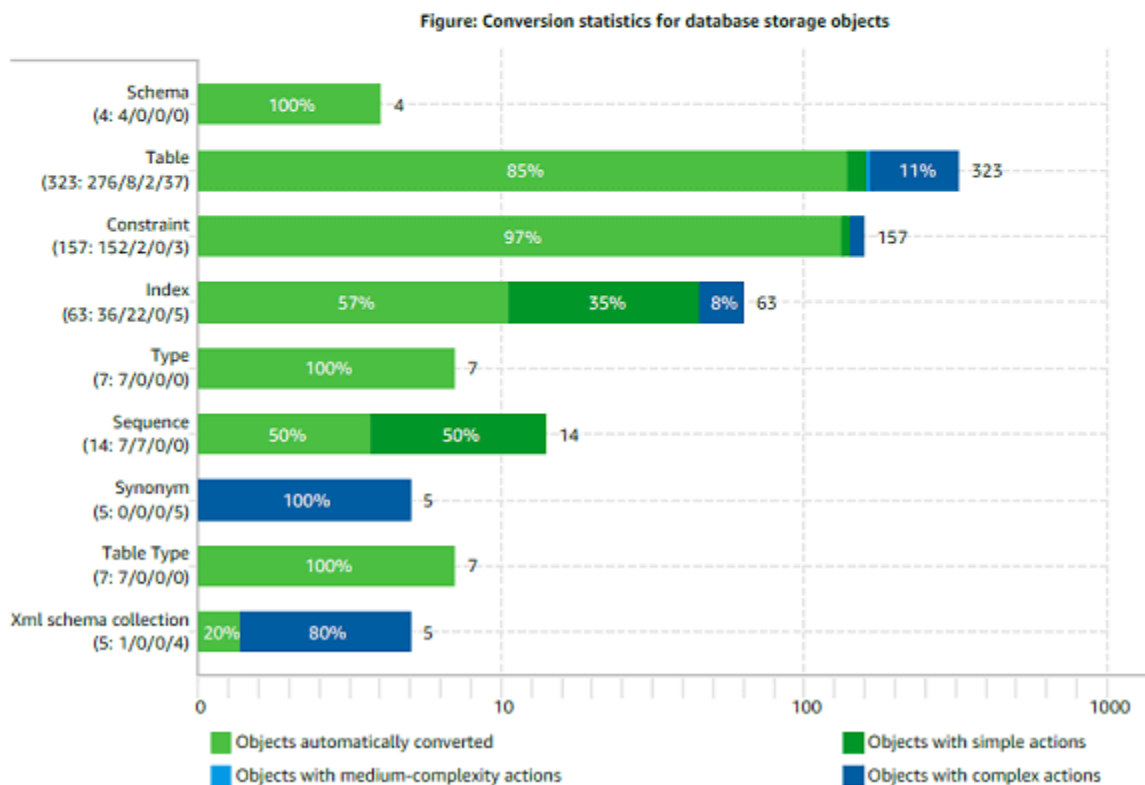
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."



Wenn Sie die Option In CSV speichern wählen, werden drei CSV-Dateien AWS SCT erstellt.

Die erste CSV-Datei enthält die folgenden Informationen zu den Aktionselementen:

- Kategorie
- Vorkommen — Der Dateiname, die Zeilennummer und die Position für das Element

- Nummer des Aktionspunkts

• Betreff

• Gruppe

• Beschreibung

• Referenzen zur Dokumentation

• Handlungsempfehlung

• Geschätzte Komplexität

Die zweite CSV-Datei enthält das `Action_Items_Summary` Suffix in ihrem Namen und enthält die Informationen über die Anzahl der Vorkommen aller Aktionselemente.

Im folgenden Beispiel geben die Werte in der Spalte Lernkurve den Aufwand an, der erforderlich ist, um einen Ansatz für die Konvertierung der einzelnen Aktionspunkte zu entwickeln. Die Werte in der Spalte Aufwand zur Konvertierung eines Vorkommens des Aktionspunkts geben den Aufwand an, der erforderlich ist, um die einzelnen Aktionspunkte zu konvertieren, wobei der entworfene Ansatz eingehalten wird. Die zur Angabe des erforderlichen Aufwandes verwendeten Werte basieren auf einer gewichteten Skala, die von niedrig (am geringsten) bis hoch (am meisten) reicht.

Schema	Action item	Number of occurrences	Learning curve efforts	Efforts to convert an occurrence of the action item
TEST.dbo	609	1	8	0.3
TEST.dbo	681	2	0.1	0.1
TEST.dbo	690	1	40	40
TEST.dbo	794	1	0	0.01
TEST.dbo	811	12	40	8
TEST.dbo	826	1	0	0.1
TEST.dbo	844	8	8	0.5
TEST.dbo	9997	3	0	0.3

Die dritte CSV-Datei enthält `Summary` ihren Namen und enthält die folgende Zusammenfassung:

- Kategorie
- Anzahl der Objekte
- Objekte automatisch konvertiert
- Objekte mit einfachen Aktionen
- Objekte mit Aktionen mittlerer Komplexität
- Objekte mit komplexen Aktionen

- Gesamtzahl der Codezeilen

Erstellung eines Multiserver-Bewertungsberichts für die Datenbankmigration

Erstellen Sie einen Bewertungsbericht für mehrere Server, um die beste Zielrichtung für Ihre gesamte Umgebung zu ermitteln.

In einem Multiserver-Bewertungsbericht werden mehrere Server anhand der Eingaben bewertet, die Sie für jede Schemadefinition, die Sie bewerten möchten, angeben. Ihre Schemadefinition enthält Datenbankserver-Verbindungsparameter und den vollständigen Namen jedes Schemas. Nach der Bewertung jedes Schemas wird ein zusammenfassender, aggregierter Bewertungsbericht für die Datenbankmigration auf Ihren mehreren Servern AWS SCT erstellt. Dieser Bericht zeigt die geschätzte Komplexität für jedes mögliche Migrationsziel.

Sie können ihn verwenden AWS SCT, um einen Multiserver-Bewertungsbericht für die folgenden Quell- und Zieldatenbanken zu erstellen.

Quelldatenbank	Zieldatenbank
Amazon Redshift	Amazon Redshift
Azure SQL Database	Aurora MySQL, MySQL, PostgreSQL, MySQL, PostgreSQL, MySQL, PostgreSQL, MySQL
Azure Synapse Analytik	Amazon Redshift
BigQuery	Amazon Redshift
Greenplum	Amazon Redshift
IBM Db2 für z/OS	Amazon Aurora MySQL-kompatible Edition (Aurora MySQL), PostgreSQL kompatible Edition (Amazon Aurora PostgreSQL), MySQL, PostgreSQL, PostgreSQL

Quelldatenbank	Zieldatenbank
IBM Db2 (LUW)	Aurora MySQL, MySQL, MySQL, PostgreSQL, MariaDB, MySQL, PostgreSQL, MariaDB, MySQL, PostgreSQL
Microsoft SQL Server	Aurora MySQL, Aurora PostgreSQL, Amazon Redshift, Babelfish für Aurora PostgreSQL, MySQL, PostgreSQL MariaDB PostgreSQL, MySQL, PostgreSQL
MySQL	Aurora PostgreSQL, MySQL, PostgreSQL
Netezza	Amazon Redshift
Oracle	Aurora MySQL, Aurora PostgreSQL, Amazon Redshift, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL	Aurora MySQL, MySQL, PostgreSQL, MySQL, PostgreSQL, MySQL, PostgreSQL, MySQL
SAP ASE	Aurora MySQL, MySQL, MySQL, PostgreSQL, MariaDB, MySQL, PostgreSQL, MariaDB, MySQL, PostgreSQL
Snowflake	Amazon Redshift
Teradata	Amazon Redshift
Vertica	Amazon Redshift

Durchführung einer Multiserver-Bewertung

Gehen Sie wie folgt vor, um eine Multiserver-Bewertung mit durchzuführen AWS SCT. Sie müssen kein neues Projekt erstellen, AWS SCT um eine Multiserveranalyse durchzuführen. Bevor Sie beginnen, stellen Sie sicher, dass Sie eine CSV-Datei (Datei mit durch Kommas getrennten Werten) mit Datenbankverbindungsparametern vorbereitet haben. Stellen Sie außerdem sicher, dass Sie alle erforderlichen Datenbanktreiber installiert haben, und legen Sie den Speicherort der Treiber

in den AWS SCT Einstellungen fest. Weitere Informationen finden Sie unter [Die erforderlichen Datenbanktreiber werden heruntergeladen](#).

Um eine Bewertung auf mehreren Servern durchzuführen und einen aggregierten zusammenfassenden Bericht zu erstellen

1. Wählen Sie in AWS SCT Datei, Neue Multiserver-Bewertung. Das Dialogfeld Neue Multiserver-Bewertung wird geöffnet.

New multiserver assessment

Enter the project name, location to store reports and project files, and location of your connections file.

Project name: Multiserver-Assessment-Project

Location: C:\AWS-SCT-Demo [Browse]

Connections file: C:\AWS-SCT-Demo\connection_example.csv [Browse]

[Download a connections file example](#)

Create AWS SCT projects for each source database

Add mapping rules to these projects and save conversion statistics for offline use

[Run] [Cancel]

2. Wählen Sie Beispiel für eine Verbindungsdatei heruntergeladen, um eine leere Vorlage einer CSV-Datei mit Datenbankverbindungsparametern herunterzuladen.
3. Geben Sie Werte für den Projektnamen, den Standort (zum Speichern von Berichten) und die Verbindungsdatei (eine CSV-Datei) ein.
4. Wählen Sie AWS SCT Projekte für jede Quelldatenbank erstellen, um nach der Generierung des Bewertungsberichts automatisch Migrationsprojekte zu erstellen.
5. Wenn die Option AWS SCT Projekte für jede Quelldatenbank erstellen aktiviert ist, können Sie „Zuordnungsregeln zu diesen Projekten hinzufügen“ wählen und Konvertierungsstatistiken für die Offline-Verwendung speichern. In diesem Fall AWS SCT werden jedem Projekt Zuordnungsregeln hinzugefügt und die Metadaten der Quelldatenbank im Projekt gespeichert. Weitere Informationen finden Sie unter [AWS SCT im Offline-Modus ausführen](#).
6. Wählen Sie Run (Ausführen) aus.

Ein Fortschrittsbalken wird angezeigt, der das Tempo der Datenbankbewertung angibt. Die Anzahl der Ziel-Engines kann sich auf die Laufzeit der Bewertung auswirken.

- Wählen Sie Ja, wenn die folgende Meldung angezeigt wird: Die vollständige Analyse aller Datenbankserver kann einige Zeit in Anspruch nehmen. Möchten Sie fortfahren?

Wenn der Multiserver-Bewertungsbericht fertig ist, erscheint ein Bildschirm, der dies anzeigt.

- Wählen Sie Bericht öffnen, um den aggregierten zusammenfassenden Bewertungsbericht anzuzeigen.

AWS SCTGeneriert standardmäßig einen aggregierten Bericht für alle Quelldatenbanken und einen detaillierten Bewertungsbericht für jeden Schemanamen in einer Quelldatenbank. Weitere Informationen finden Sie unter [Suchen und Anzeigen von Berichten](#).

Wenn die Option AWS SCTProjekte für jede Quelldatenbank erstellen aktiviert ist,AWS SCT wird für jede Quelldatenbank ein leeres Projekt erstellt. AWS SCTerstellt auch Bewertungsberichte, wie zuvor beschrieben. Nachdem Sie diese Bewertungsberichte analysiert und das Migrationsziel für jede Quelldatenbank ausgewählt haben, fügen Sie Zieldatenbanken zu diesen leeren Projekten hinzu.

Wenn die Option Zuordnungsregeln zu diesen Projekten hinzufügen und Konvertierungsstatistiken für die Offline-Verwendung speichern aktiviert ist,AWS SCT wird für jede Quelldatenbank ein Projekt erstellt. Diese Projekte enthalten die folgenden Informationen:

- Ihre Quelldatenbank und eine virtuelle Zieldatenbankplattform. Weitere Informationen finden Sie unter [Verwendung virtueller Ziele](#).
- Eine Zuordnungsregel für dieses Quell-Ziel-Paar. Weitere Informationen finden Sie unter [Mapping-Regeln erstellen](#).
- Ein Bewertungsbericht zur Datenbankmigration für dieses Quell-Ziel-Paar.
- Quellschematadaten, mit denen Sie diesesAWS SCT Projekt im Offline-Modus verwenden können. Weitere Informationen finden Sie unter [AWS SCTIm Offline-Modus ausführen](#).

Vorbereiten einer CSV-Eingabedatei

Verwenden Sie eine CSV-Datei, wie im folgenden Beispiel gezeigt, um Verbindungsparameter als Eingabe für den Multiserver-Bewertungsbericht bereitzustellen.

```
Name,Description,Secret Manager Key,Server IP,Port,Service Name,Database name,BigQuery
path,Source Engine,Schema Names,Use Windows Authentication,Login,Password,Use
SSL,Trust store,Key store,SSL authentication,Target Engines
Sales,,,192.0.2.0,1521,pdb,,,ORACLE,Q4_2021;FY_2021,,user,password,,,,,POSTGRESQL;AURORA_POSTGR
Marketing,,,ec2-a-b-c-d.eu-
west-1.compute.amazonaws.com,1433,,target_audience,,MSSQL,customers.dbo,,user,password,,,,,AURO
HR,,,192.0.2.0,1433,,employees,,MSSQL,employees.%,true,,,,,,AURORA_POSTGRESQL
Customers,,secret-name,,,,,MYSQL,customers,,,,,,AURORA_POSTGRESQL
Analytics,,,198.51.100.0,8195,,STATISTICS,,DB2LUW,BI_REPORTS,,user,password,,,,,POSTGRESQL
Products,,,203.0.113.0,8194,,,,,TERADATA,new_products,,user,password,,,,,REDSHIFT
```

Im vorherigen Beispiel wird ein Semikolon verwendet, um die beiden Schemanamen für dieSales Datenbank zu trennen. Es verwendet auch ein Semikolon, um die beiden Zieldatenbank-Migrationsplattformen für dieSales Datenbank zu trennen.

Im vorherigen Beispiel wird außerdem dieAWS Secrets Manager Verbindung mit derCustomers Datenbank und die Windows-Authentifizierung verwendet, um eine Verbindung mit derHR Datenbank herzustellen.

Sie können eine neue CSV-Datei erstellen oder eine Vorlage für eine CSV-Datei von herunterladenAWS SCT und die erforderlichen Informationen eingeben. Stellen Sie sicher, dass die erste Zeile Ihrer CSV-Datei dieselben Spaltennamen wie im vorherigen Beispiel enthält.


Um eine Vorlage der CSV-Eingabedatei herunterzuladen

1. Starten AWS SCT.
2. Wählen Sie „Datei“ und anschließend „Neue Multiserver-Bewertung“.
3. Wählen Sie Beispiel für eine Verbindungsdatei herunterladen.

Stellen Sie sicher, dass Ihre CSV-Datei die folgenden Werte enthält, die in der Vorlage bereitgestellt werden:

- Name — Die Textbeschriftung, mit der Ihre Datenbank identifiziert werden kann. AWS SCTzeigt diese Textbezeichnung im Bewertungsbericht an.
- Beschreibung — Ein optionaler Wert, für den Sie zusätzliche Informationen zur Datenbank angeben können.
- Secret Manager Key — Der Name des geheimen Codes, in dem Ihre Datenbankanmeldeinformationen gespeichert sindAWS Secrets Manager. Um Secrets Manager

zu verwenden, stellen Sie sicher, dass Sie AWS Profile darin speichern AWS SCT. Weitere Informationen finden Sie unter [Verwenden von AWS Secrets Manager](#).

 **Wichtig**

AWS SCT ignoriert den Secret Manager-Schlüsselparameter, wenn Sie die Parameter Server-IP, Port, Login und Passwort in die Eingabedatei aufnehmen.

- **Server-IP** — Der DNS-Name (Domain Service) oder die IP-Adresse Ihres Quelldatenbankservers.
- **Port** — Der Port, der für die Verbindung zu Ihrem Quelldatenbankserver verwendet wird.
- **Dienstname** — Wenn Sie einen Dienstnamen verwenden, um eine Verbindung zu Ihrer Oracle-Datenbank herzustellen, der Name des Oracle-Dienstes, zu dem eine Verbindung hergestellt werden soll.
- **Datenbankname** — Der Datenbankname. Verwenden Sie für Oracle-Datenbanken die Oracle-System-ID (SID).
- **BigQuery Pfad** — der Pfad zur Schlüsseldatei des Dienstkontos für Ihre BigQuery Quelldatenbank. Weitere Informationen zum Erstellen dieser Datei finden Sie unter [Rechte für BigQuery als Quelle](#).
- **Source Engine** — Der Typ Ihrer Quelldatenbank. Verwenden Sie einen der folgenden Werte:
 - **AZURE_MSSQL** für eine Azure SQL-Datenbank.
 - **AZURE_SYMAPSE** für eine Azure Synapse Analytics-Datenbank.
 - **GOOGLE_BIGQUERY** für eine BigQuery Datenbank.
 - **DB2ZOS** für eine IBM Db2 for z/OS-Datenbank.
 - **DB2LUW** für eine IBM Db2 LUW-Datenbank.
 - **GREENPLUM** für eine Greenplum-Datenbank.
 - **MSSQL** für eine Microsoft SQL Server-Datenbank.
 - **MYSQL** für eine MySQL-Datenbank.
 - **NETEZZA** für eine Netezza-Datenbank.
 - **ORACLE** für eine Oracle-Datenbank.
 - **POSTGRES** für eine PostgreSQL-Datenbank.
 - **REDSHIFT** für eine Amazon-Redshift-Datenbank.
 - **SNOWFLAKE** für eine Snowflake-Datenbank.
 - **SYBASE_ASE** für eine SAP ASE-Datenbank.
 - **TERADATA** für eine Teradata-Datenbank.

- VERTICA für eine Vertica-Datenbank.
- Schemanamen — Die Namen der Datenbankschemas, die in den Bewertungsbericht aufgenommen werden sollen.

Verwenden Sie für Azure SQL Database, Azure Synapse Analytics BigQuery, Netezza, SAP ASE, Snowflake und SQL Server das folgende Format des Schemanamens:

db_name.schema_name

db_name Ersetzen Sie durch den Namen der Quelldatenbank.

schema_name Ersetzen Sie durch den Namen des Quellschemas.

Fügen Sie Datenbank- oder Schemanamen ein, die einen Punkt in doppelten Anführungszeichen enthalten, wie im Folgenden gezeigt: "database.name"."schema.name".

Trennen Sie mehrere Schemanamen, indem Sie Semikolons verwenden, wie im Folgenden gezeigt: Schema1; Schema2.

Bei den Datenbank- und Schemanamen muss die Groß- und Kleinschreibung beachtet werden.

Verwenden Sie das Prozent (%) als Platzhalter, um eine beliebige Anzahl von Symbolen im Datenbank- oder Schemanamen zu ersetzen. Im vorherigen Beispiel wird der Prozentsatz (%) als Platzhalter verwendet, um alle Schemas aus der employees Datenbank in den Bewertungsbericht aufzunehmen.

- Windows-Authentifizierung verwenden — Wenn Sie die Windows-Authentifizierung verwenden, um eine Verbindung zu Ihrer Microsoft SQL Server-Datenbank herzustellen, geben Sie true ein. Weitere Informationen finden Sie unter [Verwenden der Windows-Authentifizierung bei Verwendung von Microsoft SQL Server als Quelle](#).
- Login — Der Benutzername für die Verbindung zu Ihrem Quelldatenbankserver.
- Passwort — Das Passwort für die Verbindung zu Ihrem Quelldatenbankserver.
- SSL verwenden — Wenn Sie Secure Sockets Layer (SSL) verwenden, um eine Verbindung zu Ihrer Quelldatenbank herzustellen, geben Sie true ein.
- Trust Store — Der Trust Store, der für Ihre SSL-Verbindung verwendet werden soll.
- Schlüsselspeicher — Der Schlüsselspeicher, der für Ihre SSL-Verbindung verwendet werden soll.
- SSL-Authentifizierung — Wenn Sie die SSL-Authentifizierung per Zertifikat verwenden, geben Sie true ein.

- Target Engines — Die Zieldatenbankplattformen. Verwenden Sie die folgenden Werte, um ein oder mehrere Ziele im Bewertungsbericht anzugeben:
 - AURORA_MYSQL für eine Aurora MySQL-kompatible Datenbank.
 - AURORA_POSTGRESQL für eine Aurora PostgreSQL-kompatible Datenbank.
 - BABELFISH für eine Babelfish for Aurora PostgreSQL-Datenbank.
 - MARIA_DB für eine MariaDB-Datenbank.
 - MSSQL für eine Microsoft SQL Server-Datenbank.
 - MYSQL für eine MySQL-Datenbank.
 - ORACLE für eine Oracle-Datenbank.
 - POSTGRESQL für eine PostgreSQL-Datenbank.
 - REDSHIFT für eine Amazon-Redshift-Datenbank.

Trennen Sie mehrere Ziele, indem Sie Semikolons wie folgt verwenden:MYSQL ; MARIA_DB. Die Anzahl der Ziele wirkt sich auf die Zeit aus, die für die Durchführung der Bewertung benötigt wird.

Suchen und Anzeigen von Berichten

Die Multiserver-Bewertung generiert zwei Arten von Berichten:

- Ein aggregierter Bericht aller Quelldatenbanken.
- Ein detaillierter Bewertungsbericht der Zieldatenbanken für jeden Schemanamen in einer Quelldatenbank.

Berichte werden in dem Verzeichnis gespeichert, das Sie im Dialogfeld Neue Multiserver-Bewertung als Speicherort ausgewählt haben.

Um auf die detaillierten Berichte zuzugreifen, können Sie in den Unterverzeichnissen navigieren, die nach Quelldatenbank, Schemanamen und Zieldatenbank-Engine organisiert sind.

Aggregierte Berichte enthalten in vier Spalten Informationen zur Konvertierungskomplexität einer Zieldatenbank. Die Spalten enthalten Informationen zur Konvertierung von Codeobjekten, Speicherobjekten, Syntaxelementen und zur Konvertierungskomplexität.

Das folgende Beispiel zeigt Informationen für die Konvertierung von zwei Oracle-Datenbankschemas in Amazon RDS for PostgreSQL.

Server IP address and port	Secret Manager key	Name	Description	Database name	Schema name	Code object conversion % for "Amazon RDS for PostgreSQL"	Storage object conversion % for "Amazon RDS for PostgreSQL"	Syntax Elements conversion % for "Amazon RDS for PostgreSQL"	Conversion Complexity for "Amazon RDS for PostgreSQL"
192.0.2.0:1521		Sales		ORCL	Q4_2021	97.78%	100.00%	98.76%	1
192.0.2.0:1521		Sales		pdb	FY_2021	82.35%	85.19%	99.24%	10

Dieselben vier Spalten werden an die Berichte für jede weitere angegebene Zieldatenbank-Engine angehängt.

Details zum Lesen dieser Informationen finden Sie im Folgenden.

Ergebnis für einen aggregierten Bewertungsbericht

Der aggregierte Bewertungsbericht zur Multiserver-Datenbankmigration in AWS Schema Conversion Tool ist eine CSV-Datei mit den folgenden Spalten:

- Server IP address and port
- Secret Manager key
- Name
- Description
- Database name
- Schema name
- Code object conversion % for *target_database*
- Storage object conversion % for *target_database*
- Syntax elements conversion % for *target_database*
- Conversion complexity for *target_database*

AWS SCT führt zum Sammeln von Informationen vollständige Bewertungsberichte aus und fasst die Berichte dann nach Schemata zusammen.

In dem Bericht wird in den folgenden drei Feldern der Prozentsatz der möglichen automatischen Konvertierung auf der Grundlage der Bewertung angezeigt:

Konvertierung von Codeobjekten%

Der Prozentsatz der Codeobjekte im Schema, die automatisch oder mit minimalen Änderungen konvertiert werden können. Codeobjekte umfassen Prozeduren, Funktionen, Ansichten und ähnliches.

Konvertierung von Speicherobjekten in%

Der Prozentsatz der Speicherobjekte, die SCT automatisch oder mit minimalen Änderungen konvertieren kann. Zu den Speicherobjekten gehören Tabellen, Indizes, Einschränkungen und ähnliches.

Konvertierung von Syntaxelementen%

Der Prozentsatz der Syntaxelemente, die SCT automatisch konvertieren kann. Zu den Syntaxelementen gehören `SELECTFROMDELETE`, `JOIN` Klauseln und ähnliches.

Die Berechnung der Konvertierungskomplexität basiert auf dem Konzept der Aktionspunkte. Ein Aktionselement steht für ein Problem im Quellcode, das Sie während der Migration zu einem bestimmten Ziel manuell beheben müssen. Ein Aktionselement kann mehrfach vorkommen.

Eine gewichtete Skala gibt den Grad der Komplexität für die Durchführung einer Migration an. Die Zahl 1 steht für die niedrigste Komplexitätsstufe und die Zahl 10 für die höchste Komplexitätsstufe.

Konvertieren von Datenbankschemas mithilfe von AWS SCT

Mit dem AWS Schema Conversion Tool (AWS SCT) können Sie Ihre vorhandenen Datenbankschemata von einer Datenbank-Engine in eine andere konvertieren. Das Konvertieren einer Datenbank mithilfe der AWS SCT Benutzeroberfläche kann ziemlich einfach sein, aber es gibt mehrere Dinge zu beachten, bevor Sie die Konvertierung durchführen.

Sie können zum Beispiel Folgendes verwenden AWS SCT:

- Sie können AWS SCT verwenden, um ein vorhandenes lokales Datenbankschema in eine Amazon RDS-DB-Instance zu kopieren, auf der dieselbe Engine ausgeführt wird. Mit dieser Funktion können Sie potenzielle Kosteneinsparungen kalkulieren, die beim Wechsel in die Cloud und einer Änderung des Lizenztyps möglich sind.
- In einigen Fällen können Datenbankfunktionen nicht in gleichwertige Amazon RDS-Funktionen umgewandelt werden. Wenn Sie eine Datenbank auf der Amazon Elastic Compute Cloud (Amazon EC2) -Plattform hosten und selbst verwalten, können Sie diese Funktionen emulieren, indem Sie sie durch Dienste ersetzen AWS.
- AWS SCT automatisiert einen Großteil des Prozesses der Konvertierung Ihres OLTP-Datenbankschemas (Online Transaction Processing) in eine Amazon Relational Database Service (Amazon RDS) MySQL-DB-Instance, einen Amazon Aurora-DB-Cluster oder eine PostgreSQL-DB-Instance. Die Quell- und Zieldatenbank-Engines enthalten viele verschiedene Funktionen und Fähigkeiten und AWS SCT versuchen, wo immer möglich, ein gleichwertiges Schema in Ihrer Amazon RDS-DB-Instance zu erstellen. Falls keine direkte Konvertierung möglich ist, stellt AWS SCT eine Liste mit möglichen Aktionen für Sie zusammen.

Themen

- [Erstellen von Migrationsregeln in AWS SCT](#)
- [Konvertierung Ihres Schemas mithilfe von AWS SCT](#)
- [Bearbeitung manueller Konvertierungen in AWS SCT](#)
- [Aktualisierung und Aktualisierung Ihres konvertierten Schemas in AWS SCT](#)
- [Speichern und Anwenden Ihres konvertierten Schemas in AWS SCT](#)
- [Datenbankschemas vergleichen](#)
- [Finden verwandter transformierter Objekte](#)

AWS SCT unterstützt die folgenden OLTP-Konvertierungen (Online Transaction Processing).

Quelldatenbank	Zieldatenbank
IBM Db2 für z/OS (Version 12)	Amazon Aurora MySQL-kompatible Ausgabe, Amazon Aurora PostgreSQL-kompatible Ausgabe, MySQL, PostgreSQL
IBM Db2 LUW (Versionen 9.1, 9.5, 9.7, 10.5, 11.1 und 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Microsoft Azure SQL-Datenbank	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
Microsoft SQL Server (Version 2008 R2 und höher)	Aurora MySQL, Aurora PostgreSQL, Babelfish für Aurora PostgreSQL, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL (Version 5.5 und höher)	Aurora PostgreSQL, MySQL, PostgreSQL Sie können Schema und Daten von MySQL zu einem Aurora MySQL-DB-Cluster migrieren , ohne es zu verwenden AWS SCT. Weitere Informationen finden Sie unter Migrieren von Daten zu einem Amazon Aurora-DB-Cluster .
Oracle (Version 10.2 und höher)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL (Version 9.1 und höher)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
SAP ASE (12,5, 15,0, 15,5, 15,7 und 16,0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL

Hinweise zur Konvertierung eines Data Warehouse-Schemas finden Sie unter [Konvertierung von Data Warehouse-Schemas in Amazon Redshift mithilfe von AWS SCT](#).

Um Ihr Datenbankschema in Amazon RDS zu konvertieren, gehen Sie wie folgt vor:

- [Erstellen von Migrationsregeln in AWS SCT](#)— Bevor Sie Ihr Schema mit konvertierenAWS SCT, können Sie Regeln einrichten, die den Datentyp von Spalten ändern, Objekte von einem Schema in ein anderes verschieben und die Namen von Objekten ändern.
- [Konvertierung Ihres Schemas mithilfe von AWS SCT](#)— AWS SCT erstellt eine lokale Version des konvertierten Schemas, die Sie überprüfen können, wendet sie jedoch erst auf Ihre Ziel-DB-Instance an, wenn Sie bereit sind.
- [Erstellung von Berichten zur Migrationsbewertung mitAWS SCT](#)— AWS SCT erstellt einen Bewertungsbericht zur Datenbankmigration, in dem die Schemaelemente aufgeführt sind, die nicht automatisch konvertiert werden können. Mithilfe dieses Berichts können Sie ermitteln, wo Sie in Ihrer Amazon RDS-DB-Instance ein Schema erstellen müssen, das mit Ihrer Quelldatenbank kompatibel ist.
- [Bearbeitung manueller Konvertierungen in AWS SCT](#)— Wenn Sie über Schemaelemente verfügen, die nicht automatisch konvertiert werden können, haben Sie zwei Möglichkeiten: Aktualisieren Sie das Quellschema und konvertieren Sie es erneut, oder Sie erstellen gleichwertige Schemaelemente in Ihrer Amazon RDS-DB-Zielinstanz.
- [Aktualisierung und Aktualisierung Ihres konvertierten Schemas in AWS SCT](#)— Sie können Ihr AWS SCT Projekt mit dem neuesten Schema aus Ihrer Quelldatenbank aktualisieren.
- [Speichern und Anwenden Ihres konvertierten Schemas in AWS SCT](#)— Wenn Sie bereit sind, müssen Sie das konvertierte Schema in Ihrem lokalen Projekt auf Ihre Amazon RDS-DB-Zielinstanz AWS SCT anwenden.

Erstellen von Migrationsregeln in AWS SCT

Bevor Sie Ihr Schema mit konvertierenAWS SCT, können Sie Migrationsregeln einrichten. Migrationsregeln in AWS SCT können Transformationen wie das Ändern des Datentyps von Spalten, das Verschieben von Objekten von einem Schema in ein anderes und das Ändern der Namen von Objekten durchführen. Nehmen wir zum Beispiel an, dass Sie in Ihrem Quellschema eine Reihe von Tabellen benannt haben `test_TABLE_NAME`. Sie können eine Regel einrichten, die das Präfix in `test_` das Präfix `demo_` im Zielschema ändert.

Note

Sie können Migrationsregeln nur für verschiedene Quell- und Zieldatenbank-Engines erstellen.

Sie können Migrationsregeln erstellen, die die folgenden Aufgaben ausführen:

- Präfix hinzufügen, entfernen oder ersetzen
- Suffix hinzufügen, entfernen oder ersetzen
- Spaltensortierung ändern
- Ändern des Datentyps
- Ändern Sie die Länge der `string` Datentypen `char` `varchar` `nvarchar`,, und
- Verschieben von Objekten
- Umbenennen von Objekten

Sie können Migrationsregeln für die folgenden Objekte erstellen:

- Datenbank
- Schema
- Tabelle
- Spalte

Migrationsregeln erstellen

Sie können Migrationsregeln erstellen und die Regeln als Teil Ihres Projekts speichern. Verwenden Sie bei geöffnetem Projekt das folgende Verfahren, um Migrationsregeln zu erstellen.

Um Migrationsregeln zu erstellen

1. Wählen Sie im Menü Ansicht die Option Mapping-Ansicht aus.
2. Wählen Sie unter Serverzuordnungen ein Paar aus Quell- und Zielservers aus.
3. Wählen Sie Neue Migrationsregel aus. Das Dialogfeld Transformationsregeln wird angezeigt.
4. Wählen Sie Add new rule aus. Der Liste der Regeln wird eine neue Zeile hinzugefügt.
5. Konfigurieren Sie Ihre Regel:
 - a. Geben Sie unter Name einen Namen für Ihre Regel ein.
 - b. Wählen Sie für For den Objekttyp aus, für den die Regel gelten soll.
 - c. Geben Sie einen Filter ein, der auf Objekte angewendet werden soll, bevor Sie die Migrationsregel anwenden. Die WHERE-Klausel wird mithilfe einer vergleichbaren Klausel

ausgewertet. Sie können den exakten Namen eingeben und ein Objekt auswählen oder Sie können ein Muster angeben und damit mehrere Objekte auswählen.

Die verfügbaren Felder für die Klausel where sind je nach Objekttyp unterschiedlich. Wenn als Objekttyp "Schema" ausgewählt wurde, ist nur das Feld für den Schemanamen verfügbar.

- d. Wählen Sie unter Aktionen den Typ der Migrationsregel aus, den Sie erstellen möchten.
- e. Geben Sie je nach Regeltyp einen oder zwei zusätzliche Werte ein. Um beispielsweise ein Objekt umzubenennen, geben Sie den neuen Namen des Objekts ein. Um ein Präfix zu ersetzen, geben Sie das alte Präfix und das neue Präfix ein.

Für die Datentypen char, varchar, nvarchar und string können Sie die Datentypplänge mithilfe des Multiplikationsoperators ändern. Beispielsweise wandelt der `%*4` Wert den `varchar(10)` Datentyp um `nvarchar(40)`.

6. Nachdem Sie Ihre Migrationsregel konfiguriert haben, wählen Sie Speichern, um Ihre Regel zu speichern. Zum Verwerfen Ihrer Änderungen können Sie auf Cancel klicken.

Transformation rules affect how the converted objects to be named on the target database. For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority. Default transformation rules are always at the top of the list and can be disabled or changed only in the [Conversion settings](#) tab. The rules can be exported to a file for later use in the DMS, but please note that AWS DMS *doesn't support* more than one transformation rule per schema level or per table level. Note, every rule might have to following status along with the corresponding color:

- ■ Successfully created enabled rule
- ■ Rule with incorrect data entered

Transformation rule: For **tables** where database name is like '%' and schema name is like '%' and table name is like 'test_%' add prefix 'demo_%'

Name: Transformation rule

For: table

where database name like: % schema name like: % table name like: test_%

Actions: add prefix demo_%

Buttons: Save, Cancel, Add new rule, Export script for DMS, Import script into SCT, Save all, Close

7. Wenn Sie mit dem Hinzufügen, Bearbeiten und Löschen der Regeln fertig sind, klicken Sie auf Save All, um alle Änderungen zu speichern.
8. Wählen Sie Schließen, um das Dialogfeld mit den Transformationsregeln zu schließen.

Sie können das Umschaltssymbol verwenden, um eine Migrationsregel zu deaktivieren, ohne sie zu löschen. Sie können das Kopiersymbol verwenden, um eine bestehende Migrationsregel zu duplizieren. Sie können das Stiftsymbol verwenden, um eine bestehende Migrationsregel zu bearbeiten. Sie können das Löschsymbolsymbol verwenden, um eine bestehende Migrationsregel zu löschen. Um alle Änderungen zu speichern, die Sie an Ihren Migrationsregeln vornehmen, wählen Sie **Alle speichern**.

Migrationsregeln exportieren

Wenn Sie Ihre Daten AWS DMS früher von Ihrer Quelldatenbank in Ihre Zieldatenbank migrieren, können Sie Informationen zu Ihren Migrationsregeln angeben AWS DMS. Weitere Informationen zu Aufgaben finden Sie unter [Arbeiten mit AWS Database Migration Service Replikationsaufgaben](#).

Um Migrationsregeln zu exportieren

1. Wählen Sie in der AWS Schema Conversion Tool im Menü Ansicht die Option Mapping-Ansicht.
2. Wählen Sie unter Migrationsregeln eine Migrationsregel aus und klicken Sie dann auf Migrationsregel ändern.
3. Wählen Sie Skript exportieren für AWS DMS.
4. Navigieren Sie zum gewünschten Speicherort des Skripts und klicken Sie auf Save. Ihre Migrationsregeln werden als JSON-Skript gespeichert, das von verwendet werden kann AWS DMS.

Konvertierung Ihres Schemas mithilfe von AWS SCT

Nachdem Sie Ihr Projekt sowohl mit Ihrer Quelldatenbank als auch mit Ihrer Amazon RDS-DB-Zielinstanz verbunden haben, zeigt Ihr AWS Schema Conversion Tool Projekt das Schema aus Ihrer Quelldatenbank im linken Bereich an. Das Schema wird in einer Strukturansicht dargestellt und jeder Knoten wird per Lazy Loading abgerufen. Wenn Sie einen Knoten in der Strukturansicht auswählen, fordert AWS SCT die Schemainformationen zu diesem Zeitpunkt aus der Quelldatenbank an.

Sie können Schemaelemente in der Quelldatenbank auswählen und dann das Schema in ein entsprechendes Schema für die DB-Engine der DB-Ziel-Instance konvertieren. Sie können beliebige Schemaelemente aus der Quelldatenbank für die Konvertierung auswählen. Falls das zu konvertierende Schemaelement von einem übergeordneten Element abhängig ist, generiert AWS SCT auch dafür ein entsprechendes Schema. Nehmen wir zum Beispiel an, Sie wählen eine

Tabelle für die Konvertierung aus. Wenn ja, AWS SCT generiert das Schema für die Tabelle und die Datenbank, in der sich die Tabelle befindet.

Konvertieren des Schemas

Um ein Schema aus Ihrer Quelldatenbank zu konvertieren, aktivieren Sie das Kontrollkästchen für den Namen des zu konvertierenden Schemas. Wählen Sie als Nächstes dieses Schema im linken Bereich Ihres Projekts aus. AWS SCT hebt den Schemanamen blau hervor. Öffnen Sie das Kontextmenü (Rechtsklick) für das Schema und wählen Sie Schema konvertieren, wie im Folgenden gezeigt.

File Actions Main view Settings Applications Help Add source Add target

Connected. Click to disconnect

Servers

- SQL Server - ec2-52-17-21-76.eu-west-1.compute.am
 - Databases [12]
 - AdventureWorks2012_CS
 - alfresco
 - GOLD_TEST_SS_PG
 - LARGE_DB_SS
 - master
 - model
 - msdb
 - tempdb
 - TEST**
 - vmap
 - vpas
 - vrecon
 - Server Objects
 - SQL Server Agent
 - Applications
 - SQL Scripts
 - noSQL Clusters
 - ETL

Create mapping...
 Create report
Convert schema
 Register agent
 Compare schema
 Load schema
 Hide schema
 Refresh from database
 Collect statistics
 Upload statistics
 Create DMS task
 Create Local & DMS task
 Create Local task
 Add virtual partitioning
 Save as SQL

Properties SQL Related converted objects Statistics

Name	
Created or last modified	
Created	2021-09-06 09:56:08.26
Object name	
Name	TEST
compatibility-level	100
collation-name	SQL_Latin1_General_CP1_CI_AS

Properties SQL Apply status Key management

Name	
Category	
Name	<Aurora_MySQL (virtual)>

Nachdem Sie das Schema von der Quelldatenbank konvertiert haben, können Sie aus dem linken Projektbereich Schemaelemente auswählen und das konvertierte Schema im mittleren Projektbereich anzeigen. Im mittleren Bereich werden unten die Eigenschaften des konvertierten Schemas und der SQL-Befehl zum Erstellen desselben wie folgt angezeigt:

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the source database structure: SQL Server - ec2-52-17-21-76.eu-west-1.cc > Databases [12] > AdventureWorks2012_CS > dbo > Tables [4] > Account. The main pane shows the SQL definition for the [Account] table in the source database:

```

1 CREATE TABLE [dbo].[Account] (
2 [ID] numeric(14,0) NOT NULL,
3 [AccountNo] varchar(16) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
4 [CurrencyID] numeric(3,0) NOT NULL,
5 [Description] varchar(160) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
6 [CustomerID] numeric(14,0) NOT NULL,
7 [StateID] numeric(2,0) NOT NULL,
8 [AccountBalance] numeric(14,3) NOT NULL,
9 [BlockedAmount] numeric(14,3) NOT NULL,
10 [Opendate] datetime NULL,
11 [Closedate] datetime NULL,
12 [RespManagerID] numeric(5,0) NULL,
13 [BankID] varchar(10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
14 )
15 ON [PRIMARY];

```

Below the source SQL, the target Amazon RDS instance is selected: Target Amazon RDS for MySQL table: Account. The converted SQL definition for the target instance is shown:

```

1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo VARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,

```

Nachdem das Schema konvertiert ist, können Sie das Projekt speichern. Die Schemainformationen von der Quelldatenbank werden zusammen mit dem Projekt gespeichert. Dank dieser Funktionalität können Sie offline ohne Verbindung zur Quelldatenbank arbeiten. AWS SCT stellt eine Verbindung zur Quelldatenbank her, um das Schema in Ihrem Projekt zu aktualisieren, wenn Sie Refresh from Database (Aktualisieren von Datenbank) für Ihre Quelldatenbank wählen. Weitere Informationen finden Sie unter [Aktualisierung und Aktualisierung Ihres konvertierten Schemas in AWS SCT](#).

Sie können einen Bewertungsbericht zur Datenbankmigration erstellen, in dem die nicht automatisch konvertierbaren Elemente aufgeführt werden. Mit diesem Bewertungsbericht können Sie die Schemaelemente ermitteln und auflösen, die nicht automatisch konvertiert werden können. Weitere Informationen finden Sie unter [Erstellung von Berichten zur Migrationsbewertung mit AWS SCT](#).

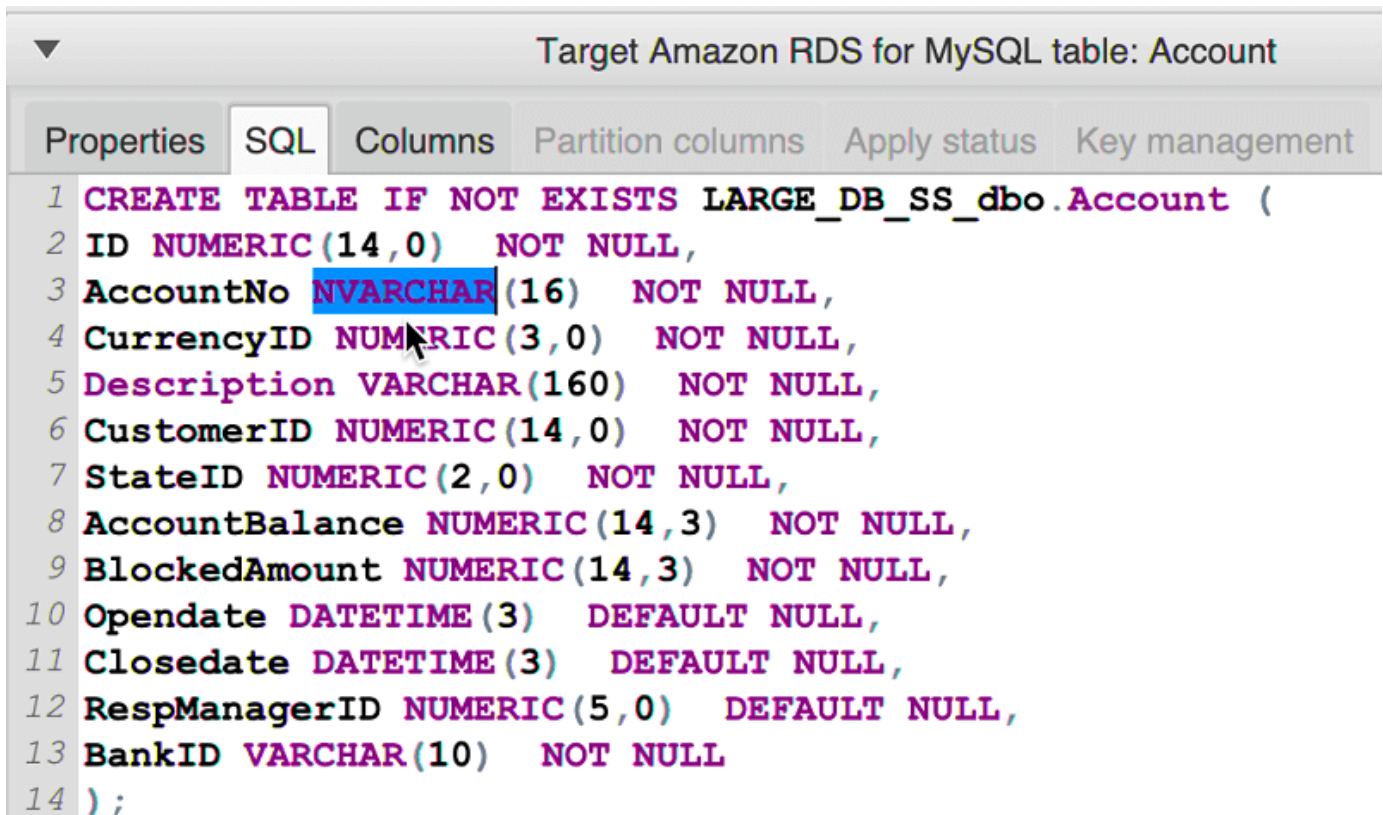
Wenn AWS SCT ein konvertiertes Schema generiert, wird dieses nicht sofort in die DB-Ziel-Instance übernommen. Stattdessen wird das konvertierte Schema lokal gespeichert, bis Sie es für die DB-Ziel-Instance übernehmen. Weitere Informationen finden Sie unter [Ihr konvertiertes Schema anwenden](#).

Konvertiertes Schema bearbeiten

Sie können ein konvertiertes Schema bearbeiten und die Änderungen als Teil des Projekts speichern.

So bearbeiten Sie ein konvertiertes Schema

1. Wählen Sie im linken Projektbereich, in dem das Schema von der Quelldatenbank angezeigt wird, das Schemaelement aus, für das Sie das konvertierte Schema ändern möchten.
2. Wählen Sie unten im mittleren Bereich, in dem das konvertierte Schema für das ausgewählte Element angezeigt wird, die Registerkarte SQL aus.
3. Nehmen Sie im Text, der unter der Registerkarte SQL angezeigt wird, die erforderlichen Änderungen am Schema vor. Das Schema wird automatisch für das Projekt gespeichert, wenn Sie es aktualisieren.



The screenshot shows the 'Target Amazon RDS for MySQL table: Account' window. The 'SQL' tab is selected, displaying the following SQL code:

```
1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (  
2 ID NUMERIC(14,0) NOT NULL,  
3 AccountNo NVARCHAR(16) NOT NULL,  
4 CurrencyID NUMERIC(3,0) NOT NULL,  
5 Description VARCHAR(160) NOT NULL,  
6 CustomerID NUMERIC(14,0) NOT NULL,  
7 StateID NUMERIC(2,0) NOT NULL,  
8 AccountBalance NUMERIC(14,3) NOT NULL,  
9 BlockedAmount NUMERIC(14,3) NOT NULL,  
10 Opendate DATETIME(3) DEFAULT NULL,  
11 Closedate DATETIME(3) DEFAULT NULL,  
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,  
13 BankID VARCHAR(10) NOT NULL  
14 );
```

Die Änderungen, die Sie am konvertierten Schema vorgenommen haben, werden für das Projekt gespeichert, wenn Sie es aktualisieren. Wenn Sie ein Schemaelement aus der Quelldatenbank neu konvertieren, aber das zuvor konvertierte Schema für dieses Element bereits aktualisiert hatten, werden die erfolgten Aktualisierungen durch das auf Basis der Quelldatenbank neu konvertierte Schemaelement ersetzt.

Löschen eines konvertierten Schemas

Bis Sie das Schema für die DB-Ziel-Instance übernehmen, speichert AWS SCT das konvertierte Schema nur lokal in Ihrem Projekt. Sie können das geplante Schema aus Ihrem Projekt löschen, indem Sie den Baumstrukturknoten für Ihre DB-Instance auswählen und dann Aus Datenbank aktualisieren wählen. Da kein Schema in Ihre Ziel-DB-Instance geschrieben wurde, werden bei der Aktualisierung aus der Datenbank die geplanten Schemaelemente in Ihrem AWS SCT Projekt entfernt, sodass sie dem entsprechen, was in Ihrer Quell-DB-Instance vorhanden ist.

Bearbeitung manueller Konvertierungen in AWS SCT

Der Bewertungsbericht enthält eine Liste von Elementen, die nicht automatisch in die Datenbank-Engine Ihrer Amazon RDS-DB-Zielinstanz konvertiert werden können. Für jedes nicht konvertierbare Element wird auf der Registerkarte Action Items ein Aktionselement angegeben.

Sie können wie folgt auf die Aktionselemente im Bewertungsbericht reagieren:

- Sie ändern das Schema der Quelldatenbank.
- Sie ändern das Schema der Zieldatenbank.

Ändern Ihres Quellschemas

Für einige Elemente ist es möglicherweise einfacher, das Datenbankschema in Ihrer Quelldatenbank in ein Schema zu ändern, das automatisch konvertiert werden kann. Vergewissern Sie sich zunächst, dass die neuen Änderungen mit der Anwendungsarchitektur kompatibel sind, und aktualisieren Sie dann das Schema der Quelldatenbank. Aktualisieren Sie abschließend das Projekt mit den neuen Schemainformationen. Anschließend können Sie das aktualisierte Schema konvertieren und den Bewertungsbericht zur Datenbankmigration erneut erstellen. Für die im Quellschema geänderten Elemente werden keine Aktionselemente mehr angezeigt.

Der Vorteil dieses Verfahrens besteht darin, dass das aktualisierte Schema immer verfügbar ist, wenn Sie von der Quelldatenbank aktualisieren.

Ihr Zielschema ändern

Bei einigen Elementen ist es einfacher, das konvertierte Schema für die Zieldatenbank zu übernehmen und anschließend für die Elemente, die nicht automatisch konvertiert werden konnten,

entsprechende Schemaelemente manuell zur Zieldatenbank hinzuzufügen. Durch Übernehmen des jeweiligen Schemas können Sie automatisch konvertierbare Schemata in die DB-Ziel-Instance schreiben. Weitere Informationen finden Sie unter [Speichern und Anwenden Ihres konvertierten Schemas in AWS SCT](#).

Das Schema, das in die DB-Ziel-Instance geschrieben wird, enthält nur automatisch konvertierbare Elemente. Nachdem Sie das Schema für die DB-Ziel-Instance übernommen haben, können Sie manuell ein (mit der Quelldatenbank) gleichwertiges Schema in der DB-Ziel-Instance erstellen. Die im Bewertungsbericht zur Datenbankmigration angegebenen Aktionselemente enthalten Vorschläge zum Erstellen eines gleichwertigen Schemas.

Warning

Wenn Sie ein Schema manuell in der DB-Ziel-Instance erstellen, sollten Sie eine Kopie davon speichern. Falls Sie das konvertierte Schema vom Projekt erneut in die DB-Ziel-Instance übernehmen, werden Ihre manuellen Änderungen überschrieben.

In einigen Fällen ist es nicht möglich, ein gleichwertiges Schema in der DB-Ziel-Instance zu erstellen. Möglicherweise muss das Design der Anwendung und der Datenbank in Teilen überarbeitet werden, damit Sie die von der DB-Engine bereitgestellte Funktionalität für Ihre DB-Ziel-Instance nutzen können. In anderen Fällen kann das Schema, das nicht automatisch konvertiert werden kann, ignoriert werden.

Aktualisierung und Aktualisierung Ihres konvertierten Schemas in AWS SCT

Sie können sowohl das Quell- als auch das Zielschema im AWS Schema Conversion Tool-Projekt aktualisieren.

- **Quelle** — Wenn Sie das Schema für Ihre Quelldatenbank aktualisieren, AWS SCT ersetzen Sie das Schema in Ihrem Projekt durch das neueste Schema aus Ihrer Quelldatenbank. Mit dieser Funktion können Sie das Projekt aktualisieren, wenn das Schema der Quelldatenbank geändert wurde.
- **Ziel** — Wenn Sie das Schema für Ihre Amazon RDS-Ziel-DB-Instance aktualisieren, AWS SCT ersetzen Sie das Schema in Ihrem Projekt durch das neueste Schema Ihrer Ziel-DB-Instance. Falls kein Schema für die DB-Ziel-Instance übernommen wurde, löscht AWS SCT das konvertierte

Schema aus dem Projekt. Anschließend können Sie das Schema von der Quelldatenbank für eine neue DB-Ziel-Instance konvertieren.

Sie aktualisieren das Schema in Ihrem AWS SCT Projekt, indem Sie Aus Datenbank aktualisieren wählen.

Note

Wenn Sie Ihr Schema aktualisieren, werden Metadaten nur dann AWS SCT geladen, wenn sie benötigt werden. Um das gesamte Schema Ihrer Datenbank vollständig zu laden, öffnen Sie das Kontextmenü (Rechtsklick) für Ihr Schema und wählen Sie Schema laden. Sie können diese Option beispielsweise verwenden, um alle Metadaten für Ihre Datenbank auf einmal zu laden und dann offline zu arbeiten.

Speichern und Anwenden Ihres konvertierten Schemas in AWS SCT

Wenn das AWS Schema Conversion Tool ein konvertiertes Schema generiert (wie in [Konvertierung Ihres Schemas mithilfe von AWS SCT](#) gezeigt), wird dieses nicht sofort für die DB-Ziel-Instance übernommen. Stattdessen wird das konvertierte Schema lokal im Projekt gespeichert, bis Sie es für die DB-Ziel-Instance übernehmen. Mit dieser Funktion können Sie mit Schemaelementen arbeiten, die sich nicht automatisch in die Zieldatenbank-Engine konvertieren lassen. Weitere Informationen zu Elementen, die nicht automatisch konvertierbar sind, finden Sie unter [Erstellung von Berichten zur Migrationsbewertung mit AWS SCT](#).

Optional können Sie das konvertierte Schema mithilfe des Tools als SQL-Skript in einer Datei speichern, bevor Sie das Schema für die DB-Ziel-Instance übernehmen. Mit diesem Tool können Sie das konvertierte Schema auch direkt für die DB-Ziel-Instance übernehmen.

Speichern Ihres konvertierten Schemas in einer Datei

Sie können das konvertierte Schema als SQL-Skripts in einer Textdatei speichern. Auf diese Weise können Sie die von AWS SCT generierten SQL-Skripts modifizieren, um solche Elemente zu berücksichtigen, die das Tool nicht automatisch konvertieren kann. Anschließend führen Sie die aktualisierten Skripts auf der DB-Ziel-Instance aus und übernehmen das konvertierte Schema in die Zieldatenbank.

Um Ihr konvertiertes Schema als SQL-Skript zu speichern

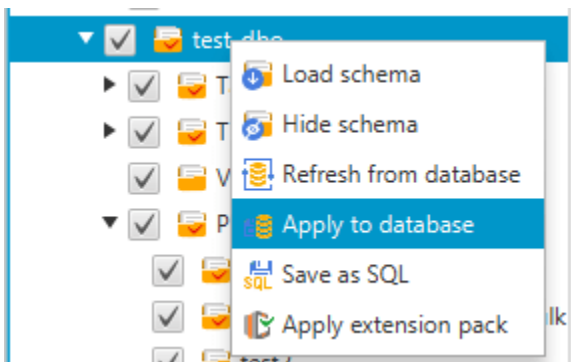
1. Wählen Sie Ihr Schema und öffnen Sie das Kontextmenü (Rechtsklick).
2. Wählen Sie Als SQL speichern.
3. Geben Sie den Namen der Datei ein und wählen Sie Speichern.
4. Speichern Sie Ihr konvertiertes Schema mit einer der folgenden Optionen:
 - Einzelne Datei
 - Eine Datei pro Stufe
 - Eine Datei pro Kontoauszug

Um das Format des SQL-Skripts auszuwählen

1. Wählen Sie im Menü Einstellungen die Option Projekteinstellungen.
2. Wählen Sie Scripts speichern.
3. Wählen Sie für Anbieter die Datenbankplattform aus.
4. Wählen Sie unter SQL-Skripts speichern in aus, wie Sie Ihr Datenbankschema-Skript speichern möchten.
5. Wählen Sie OK, um die Einstellungen zu speichern.

Ihr konvertiertes Schema anwenden

Wenn Sie bereit sind, Ihr konvertiertes Schema auf Ihre Amazon RDS-DB-Zielinstanz anzuwenden, wählen Sie das Schemaelement im rechten Bereich Ihres Projekts aus. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Schemaelement und wählen Sie Apply to database aus, wie hier gezeigt.



Das Schema des Erweiterungspakets

Wenn Sie zum ersten Mal ein konvertiertes Schema für die DB-Ziel-Instance übernehmen, fügt AWS SCT ein weiteres Schema zu Ihrer DB-Ziel-Instance hinzu. Über dieses Schema werden Systemfunktionen der Quelldatenbank implementiert, die zum Schreiben des konvertierten Schemas in die Ziel-DB-Instance benötigt werden. Das Schema wird als Erweiterungspaketschema bezeichnet.

Dieses Erweiterungspaketschema sollten Sie nicht ändern, andernfalls könnten im konvertierten Schema, das auf die DB-Ziel-Instance geschrieben wird, unerwünschte Ergebnisse auftreten. Nachdem das Schema vollständig zur DB-Ziel-Instance migriert wurde und Sie AWS SCT nicht mehr benötigen, können Sie das Erweiterungspaketschema löschen.

Das Erweiterungspaketschema wird wie folgt entsprechend Ihrer Quelldatenbank benannt:

- IBM Db2 LUW: `aws_db2_ext`
- Microsoft SQL Server: `aws_sqlserver_ext`
- MySQL: `aws_mysql_ext`
- Oracle: `aws_oracle_ext`
- PostgreSQL: `aws_postgresql_ext`
- SAP-BASIS: `aws_sapase_ext`

Weitere Informationen finden Sie unter [Verwenden Sie die AWS Lambda Funktionen aus dem AWS SCT Erweiterungspaket](#).

Datenbankschemas vergleichen

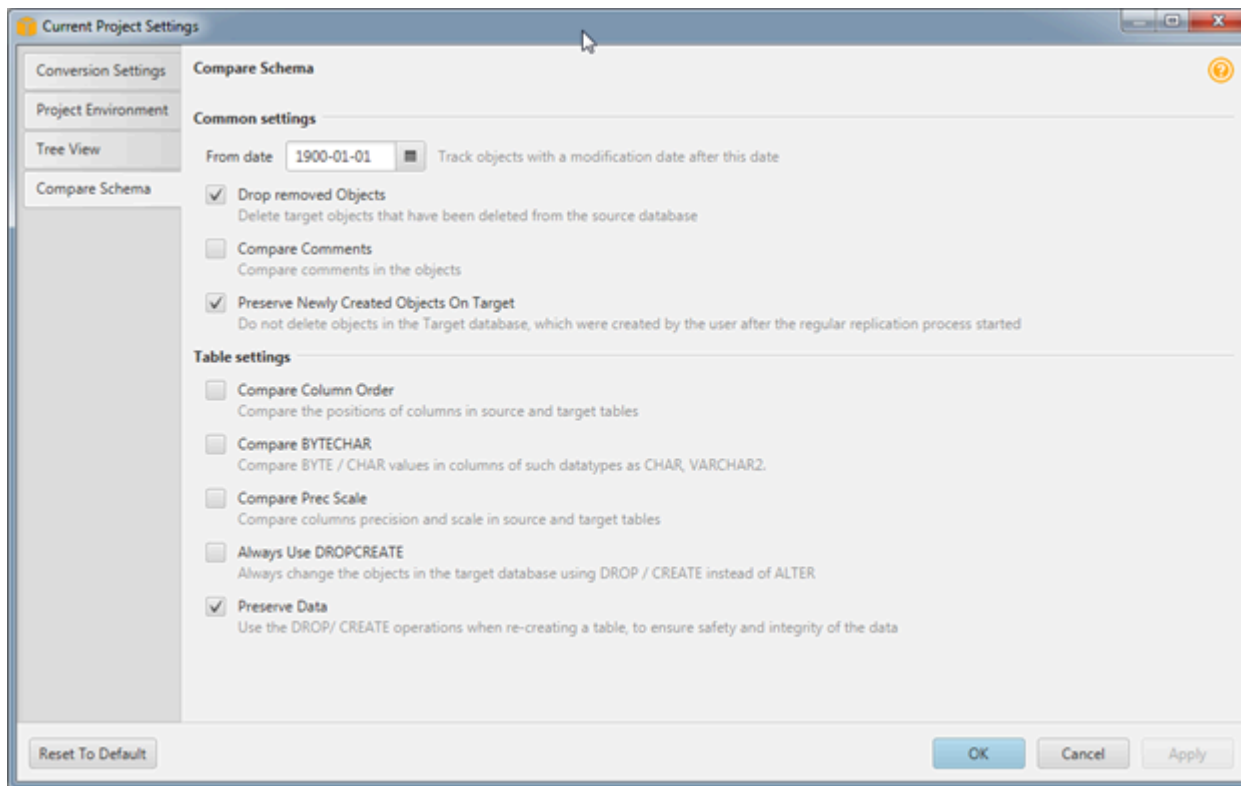
Wenn Sie nach der Migration Änderungen an Ihrem Quell- oder Zielschema durchgeführt haben, können Sie die beiden Datenbankschemata mit dem AWS SCT vergleichen. Sie können Schemata der gleichen Version vergleichen oder eine Version mit einer früheren Version des Quellschemas.

Die folgenden Schemavergleiche werden unterstützt:

- Oracle mit Oracle, Versionen 12.1.0.2.0, 11.1.0.7.0, 11.2.0.1.0 und 10
- SQL Server mit SQL Server, Versionen 2016, 2014, 2012, 2008 RD2 und 2008
- PostgreSQL to PostgreSQL and Aurora PostgreSQL-Compatible Edition, Versionen 9.6, 9.5.9, 9.5.4

- MySQL mit MySQL, Versionen 5.6.36, 5.7.17 und 5.5

Sie geben Einstellungen für den Schemavergleich auf der Registerkarte Compare Schema (Schema vergleichen) der Seite Project Settings (Projekteinstellungen) an.

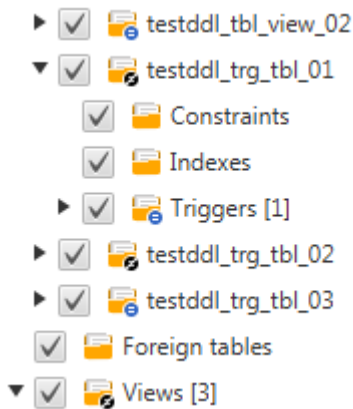


Zum Vergleichen von Schemata wählen Sie diese aus und das AWS SCT gibt die Objekte an, die sich in beiden Versionen unterscheiden sowie die, die sich gleichen.

So vergleichen Sie zwei Schemata

1. Öffnen Sie ein vorhandenes AWS SCT-Projekt oder erstellen Sie ein Projekt und stellen Sie eine Verbindung zu den Quell- und Zielpunkten her.
2. Wählen Sie das zu vergleichende Schema aus.
3. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Schema vergleichen.

AWS SCT zeigt Unterschiede zwischen den beiden Schemata an, indem ein schwarzer Kreis zum Objektsymbol hinzugefügt wird.



Sie können die Ergebnisse des Schemavergleichs auf ein einzelnes Objekt, auf eine einzelne Kategorie von Objekten oder auf das gesamte Schema anwenden. Aktivieren Sie das Kontrollkästchen neben der Kategorie, dem Objekt oder dem Schema, auf die bzw. das die Ergebnisse angewendet werden sollen.

Finden verwandter transformierter Objekte

Nach einer Schemakonvertierung hat das AWS SCT möglicherweise mehrere Objekte für ein Schemaobjekt auf der Quelldatenbank angelegt. Wenn Sie beispielsweise eine Konvertierung von Oracle zu PostgreSQL durchführen, verwendet das AWS SCT jeden Oracle-Auslöser und transformiert diesen in einen Auslöser und eine Auslöser-Funktion im PostgreSQL-Ziel. Zudem erstellt das AWS SCT, wenn es eine Oracle-Paketfunktion oder -prozedur in PostgreSQL konvertiert, eine gleichwertige Funktion sowie eine INIT-Funktion, die als INIT-Block ausgeführt werden sollte, ehe die Prozedur oder Funktion ausgeführt werden kann.

Mit dem folgenden Verfahren können Sie alle verwandten Objekte anzeigen, die während der Schemakonvertierung erstellt wurden.

So zeigen Sie verwandte Objekte an, die während der Schemakonvertierung erstellt wurden

1. Wählen Sie nach der Schemakonvertierung das konvertierte Objekt in der Zielstrukturansicht aus.
2. Wählen Sie die Registerkarte Related Converted Objects (Verwandte konvertierte Objekte) aus.
3. Sehen Sie sich die Liste der verwandten Zielobjekte an.

Konvertierung von Data Warehouse-Schemas in Amazon Redshift mithilfe von AWS SCT

Das AWS Schema Conversion Tool (AWS SCT) automatisiert einen Großteil des Prozesses der Konvertierung Ihres Data Warehouse-Schemas in ein Amazon Redshift-Datenbankschema. Da die Quell- und Zieldatenbank-Engines können viele verschiedene Features und Funktionen enthalten können, versucht AWS SCT, ein vergleichbares Schema in Ihrer Zieldatenbank zu erstellen. Falls keine direkte Konvertierung möglich ist, stellt AWS SCT einen Bewertungsbericht mit einer Liste möglicher Aktionen für Sie zusammen. Mit AWS SCT können Sie Schlüssel verwalten, Datentypen und Objekte abbilden und manuelle Umwandlungen erstellen.

AWS SCT kann die folgenden Data Warehouse-Schemas in Amazon Redshift konvertieren.

- Amazon Redshift
- Azure Synapse Analytics (Version 10)
- BigQuery
- Greenplum-Datenbank (Version 4.3)
- Microsoft SQL Server (Version 2008 und höher)
- Netezza (Version 7.0.3 und höher)
- Oracle (Version 10.2 und höher)
- Schneeflocke (Version 3)
- Teradata (Version 13 und höher)
- Vertica (Version 7.2 und höher)

Hinweise zur Konvertierung eines OLTP-Datenbankschemas (Online Transaction Processing) finden Sie unter [Konvertieren von Datenbankschemas mithilfe von AWS SCT](#).

Gehen Sie wie folgt vor, um ein Data Warehouse-Schema zu konvertieren:

1. Geben Sie die Optimierungsstrategie und die Regeln sowie die Migrationsregeln an, die Sie verwenden AWS SCT möchten. Sie können Regeln festlegen, um den Datentyp von Spalten zu ändern, Objekte von einem Schema in ein anderes zu verschieben und Objektnamen zu bearbeiten.

- In den Einstellungen können Sie Optimierungs- und Migrationsregeln festlegen. Weitere Informationen zu Optimierungsstrategien finden Sie unter [Auswahl von Optimierungsstrategien und Regeln für die Verwendung mit AWS SCT](#). Weitere Informationen zu Migrationsregeln finden Sie unter [Erstellen von Migrationsregeln in AWS SCT](#)
2. Geben Sie Statistiken zu Ihrem Quell-Data Warehouse an, sodass AWS SCT optimieren kann, wie Ihr Data Warehouse umgewandelt wird. Sie können Statistiken direkt aus der Datenbank erfassen oder eine vorhandene Statistikdatei hochladen. Weitere Informationen über die Bereitstellung von Data-Warehouse-Statistiken finden Sie unter [Sammeln oder Hochladen von Statistiken für AWS SCT](#).
 3. Erstellen Sie einen Bewertungsbericht zur Datenbankmigration, in dem die Schemaelemente aufgeführt werden, die sich nicht automatisch umwandeln lassen. Mithilfe dieses Berichts können Sie ermitteln, wo Sie ein Schema in der Zieldatenbank manuell erstellen müssen, das kompatibel mit der Quelldatenbank ist. Weitere Informationen zum Bewertungsbericht finden Sie unter [Erstellung von Berichten zur Migrationsbewertung mit AWS SCT](#).
 4. Konvertieren Sie das Schema. AWS SCT erstellt eine lokale Version des konvertierten Schemas, damit Sie es prüfen können. Das Schema wird erst für die Zieldatenbank übernommen, wenn Sie mit der Überprüfung fertig sind. Weitere Informationen zur Umwandlung finden Sie unter [Konvertierung Ihres Schemas mit AWS SCT](#).
 5. Nachdem Sie das Schema konvertiert haben, können Sie Ihre Schlüssel verwalten und bearbeiten. Die Schlüsselverwaltung spielt eine zentrale Rolle bei der Data Warehouse-Konvertierung. Weitere Informationen zum Verwaltung von Schlüsseln finden Sie unter [Verwaltung und Anpassung von Schlüsseln in AWS SCT](#).
 6. Falls Schemaelemente vorhanden sind, die nicht automatisch konvertiert werden können, haben Sie zwei Möglichkeiten: Entweder aktualisieren Sie das Quellschema und führen die Konvertierung erneut aus oder Sie erstellen entsprechende Schemaelemente in der Zieldatenbank. Weitere Informationen zum manuellen Umwandeln von Schemaelementen finden Sie unter [Bearbeitung manueller Konvertierungen in AWS SCT](#). Weitere Informationen zur Aktualisierung Ihres Quellschemas finden Sie unter [Aktualisierung und Aktualisierung Ihres konvertierten Schemas in AWS SCT](#).
 7. Wenn Sie bereit sind, können Sie das umgewandelte Schema auf Ihre Zieldatenbank anwenden. Weitere Informationen zum Speichern und Anwenden des umgewandelten Schemas finden Sie unter [Speichern und Anwenden Ihres konvertierten Schemas in AWS SCT](#).

Berechtigungen für Amazon Redshift als Ziel

Die für Amazon Redshift als Ziel erforderlichen Berechtigungen sind im Folgenden aufgeführt:

- CREATE ON DATABASE — ermöglicht das Erstellen neuer Schemas in der Datenbank.
- CREATE ON SCHEMA — ermöglicht das Erstellen von Objekten im Datenbankschema.
- GRANT USAGE ON LANGUAGE — ermöglicht die Erstellung neuer Funktionen und Prozeduren in der Datenbank.
- GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog — stellt dem Benutzer Systeminformationen über den Amazon Redshift-Cluster zur Verfügung.
- GRANT SELECT ON pg_class_info — stellt dem Benutzer Informationen über den Stil der Tabellenverteilung zur Verfügung.

Sie können das folgende Codebeispiel verwenden, um einen Datenbankbenutzer zu erstellen und die Berechtigungen zu gewähren.

```
CREATE USER user_name PASSWORD your_password;  
GRANT CREATE ON DATABASE db_name TO user_name;  
GRANT CREATE ON SCHEMA schema_name TO user_name;  
GRANT USAGE ON LANGUAGE plpythonu TO user_name;  
GRANT USAGE ON LANGUAGE plpgsql TO user_name;  
GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog TO user_name;  
GRANT SELECT ON pg_class_info TO user_name;  
GRANT SELECT ON sys_serverless_usage TO user_name;  
GRANT SELECT ON pg_database_info TO user_name;  
GRANT SELECT ON pg_statistic TO user_name;
```

Ersetzen Sie im obigen Beispiel *user_name* durch den Namen Ihres Benutzers. Ersetzen Sie dann *db_name* durch den Namen Ihrer Amazon Redshift-Zieldatenbank. Als Nächstes ersetzen Sie *schema_name* durch den Namen Ihres Amazon Redshift-Schemas. Wiederholen Sie den GRANT CREATE ON SCHEMA Vorgang für jedes Zielschema, in dem Sie den konvertierten Code anwenden oder Daten migrieren möchten. Ersetzen Sie *abschließend your_password* durch ein sicheres Passwort.

Sie können ein Erweiterungspaket auf Ihre Amazon Redshift-Zieldatenbank anwenden. Ein Erweiterungspaket ist ein Zusatzmodul, das Quelldatenbankfunktionen emuliert, die für die Konvertierung von Objekten in Amazon Redshift erforderlich sind. Weitere Informationen finden Sie unter [Verwenden von AWS SCT Erweiterungspaketen](#).

Für diesen Vorgang AWS SCT benötigen Sie die Erlaubnis, in Ihrem Namen auf den Amazon S3-Bucket zuzugreifen. Um diese Berechtigung zu gewähren, erstellen Sie einen AWS Identity and Access Management (IAM-) Benutzer mit der folgenden Richtlinie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "arn:aws:s3:::aws-sct-*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets"
      ],
      "Resource": ""
    }
  ]
}
```

Auswahl von Optimierungsstrategien und Regeln für die Verwendung mit AWS SCT

Zum Optimieren der Konvertierung Ihres Data Warehouse-Schemas mit dem AWS Schema Conversion Tool können Sie Regeln und Strategien für das Tool festlegen. Nachdem Sie Ihr Schema konvertiert und die empfohlenen Schlüssel überprüft haben, können Sie Ihre Regeln anpassen oder die Strategie ändern, um die gewünschten Ergebnisse zu erzielen.

So legen Sie Optimierungsstrategien und -regeln fest

1. Wählen Sie Settings (Einstellungen) und anschließend Project Settings (Projekteinstellungen) aus. Das Dialogfeld Current project settings wird geöffnet.
2. Wählen Sie im linken Bereich Optimization Strategies aus. Die Optimierungsstrategien werden im rechten Bereich mit ihren Standardeinstellungen angezeigt.
3. Wählen Sie für Strategy Sector die gewünschte Optimierungsstrategie aus. Sie können aus den folgenden Optionen auswählen:
 - Metadaten verwenden, statistische Informationen ignorieren — Bei dieser Strategie werden nur Informationen aus den Metadaten für Optimierungsentscheidungen verwendet. Beispiel: Gibt es mehr als einen Index für eine Quelltable, wird die Sortierreihenfolge der Quelldatenbank verwendet und der erste Index wird zum Verteilungsschlüssel.
 - Metadaten ignorieren, statistische Informationen verwenden — Bei dieser Strategie werden Optimierungsentscheidungen ausschließlich aus statistischen Informationen abgeleitet. Diese Strategie gilt nur für Tabellen und Spalten, für die Statistiken bereitgestellt werden. Weitere Informationen finden Sie unter [Sammeln oder Hochladen von Statistiken für AWS SCT](#).
 - Verwenden Sie Metadaten und verwenden Sie statistische Informationen — In dieser Strategie werden sowohl Metadaten als auch Statistiken für Optimierungsentscheidungen verwendet.
4. Nachdem Sie die Optimierungsstrategie festgelegt haben, können Sie die zu verwendenden Regeln auswählen. Sie können aus den folgenden Optionen auswählen:
 - Choose Distribution Key and Sort Keys using metadata
 - Choose fact table and appropriate dimension for collation
 - Analyze cardinality of indexes' columns
 - Finden Sie die am häufigsten verwendeten Tabellen und Spalten in der Abfrageprotokolltable

Für jede Regel können Sie eine Gewichtung für den Sortierschlüssel und eine Gewichtung für den Verteilungsschlüssel eingeben. AWS SCT nutzt diese Gewichtungen beim Konvertieren Ihres Schemas. Wenn Sie später die empfohlenen Schlüssel überprüfen, können Sie, wenn Sie mit den Ergebnissen nicht zufrieden sind, hierher zurückkehren, um Ihre Einstellungen zu

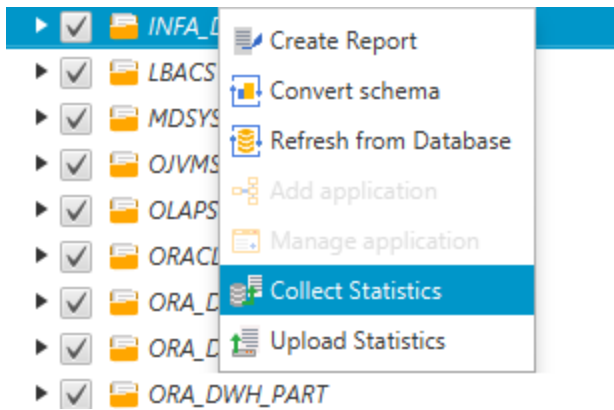
ändern. Weitere Informationen finden Sie unter [Verwaltung und Anpassung von Schlüsseln in AWS SCT](#).

Sammeln oder Hochladen von Statistiken für AWS SCT

Zum Optimieren der Konvertierung Ihres Data Warehouse-Schemas mit dem AWS Schema Conversion Tool können Sie Statistiken aus der Quelldatenbank bereitstellen, die das Tool nutzen kann. Sie können Statistiken direkt aus der Datenbank erfassen oder eine vorhandene Statistikdatei hochladen.

So stellen Sie Statistiken bereit und überprüfen diese

1. Öffnen Sie Ihr Projekt, und stellen Sie eine Verbindung zu Ihrer Quell-Datenbank her.
2. Wählen Sie ein Schemaobjekt aus dem linken Bereich Ihres Projekts aus und öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt. Wählen Sie, wie im Folgenden gezeigt, entweder Collect Statistics oder Upload Statistics aus.



3. Wählen ein Schemaobjekt aus dem linken Bereich Ihres Projekts und anschließend die Registerkarte Statistics aus. Sie können die statistischen Daten für das Objekt überprüfen.

Column Name	Stats Collection Date	Stats collection mode	Stats usage count	Stats cardinality
PART_ID	2016-06-14 15:41:23	online		9000
ADJUSTER_ID	2016-06-14 15:41:23	online		24
SPEC_ID	2016-06-14 15:41:23	online		111

Wenn Sie später die empfohlenen Schlüssel überprüfen, können Sie, wenn Sie mit den Ergebnissen nicht zufrieden sind, zusätzliche Statistiken erfassen und diesen Prozess wiederholen. Weitere Informationen finden Sie unter [Verwaltung und Anpassung von Schlüsseln in AWS SCT](#).

Erstellen von Migrationsregeln in AWS SCT

Bevor Sie Ihr Schema mit konvertieren AWS SCT, können Sie Migrationsregeln einrichten. Mit Migrationsregeln können Sie beispielsweise den Datentyp von Spalten ändern, Objekte von einem Schema in ein anderes verschieben und die Namen von Objekten ändern. Nehmen wir zum Beispiel an, dass Sie in Ihrem Quellschema eine Reihe von Tabellen benannt haben `test_TABLE_NAME`. Sie können eine Regel einrichten, die das Präfix in `test_` das Präfix `demo_` im Zielschema ändert.

Note

Sie können Migrationsregeln nur für verschiedene Quell- und Zieldatenbank-Engines erstellen.

Sie können Migrationsregeln erstellen, die die folgenden Aufgaben ausführen:

- Präfix hinzufügen, entfernen oder ersetzen
- Suffix hinzufügen, entfernen oder ersetzen
- Spaltensortierung ändern
- Ändern des Datentyps
- Ändern Sie die Länge der `string` Datentypen `char` `varchar` `nvarchar`,, und
- Verschieben von Objekten
- Umbenennen von Objekten

Sie können Migrationsregeln für die folgenden Objekte erstellen:

- Datenbank
- Schema
- Tabelle
- Spalte

Migrationsregeln erstellen

Sie können Migrationsregeln erstellen und die Regeln als Teil Ihres Projekts speichern. Verwenden Sie bei geöffnetem Projekt das folgende Verfahren, um Migrationsregeln zu erstellen.

Um Migrationsregeln zu erstellen

1. Wählen Sie im Menü Ansicht die Option Mapping-Ansicht.
2. Wählen Sie unter Serverzuordnungen ein Paar aus Quell- und Zielservers aus.
3. Wählen Sie Neue Migrationsregel aus. Das Dialogfeld Transformationsregeln wird angezeigt.
4. Wählen Sie Add new rule aus. Der Liste der Regeln wird eine neue Zeile hinzugefügt.
5. Konfigurieren Sie Ihre Regel:
 - a. Geben Sie unter Name einen Namen für Ihre Regel ein.
 - b. Wählen Sie für For den Objekttyp aus, für den die Regel gelten soll.
 - c. Geben Sie einen Filter ein, der auf Objekte angewendet werden soll, bevor Sie die Migrationsregel anwenden. Die WHERE-Klausel wird mithilfe einer vergleichbaren Klausel

ausgewertet. Sie können den exakten Namen eingeben und ein Objekt auswählen oder Sie können ein Muster angeben und damit mehrere Objekte auswählen.

Die verfügbaren Felder für die Klausel where sind je nach Objekttyp unterschiedlich. Wenn als Objekttyp "Schema" ausgewählt wurde, ist nur das Feld für den Schemanamen verfügbar.

- d. Wählen Sie unter Aktionen den Typ der Migrationsregel aus, den Sie erstellen möchten.
 - e. Geben Sie je nach Regeltyp einen oder zwei zusätzliche Werte ein. Um beispielsweise ein Objekt umzubenennen, geben Sie den neuen Namen des Objekts ein. Um ein Präfix zu ersetzen, geben Sie das alte Präfix und das neue Präfix ein.
6. Nachdem Sie Ihre Migrationsregel konfiguriert haben, wählen Sie Speichern, um Ihre Regel zu speichern. Zum Verwerfen Ihrer Änderungen können Sie auf Cancel klicken.

Transformation rules

Transformation rules affect how the converted objects to be named on the target database. For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority. Default transformation rules are always at the top of the list and can be disabled or changed only in the [Conversion settings](#) tab. The rules can be exported to a file for later use in the DMS, but please note that AWS DMS [doesn't support](#) more than one transformation rule per schema level or per table level. Note, every rule might have to following status along with the corresponding color:

- Successfully created enabled rule
- Rule with incorrect data entered

Transformation rule: For **tables** where database name is like '%' and schema name is like '%' and table name is like 'test_%' add prefix 'demo_%'

Name: Transformation rule

For: table

where database name like: % schema name like: % table name like: test_%

Actions: add prefix demo_%

Buttons: Save, Cancel, Add new rule, Save all, Close, Export script for DMS, Import script into SCT

7. Wenn Sie mit dem Hinzufügen, Bearbeiten und Löschen der Regeln fertig sind, klicken Sie auf Save All, um alle Änderungen zu speichern.
8. Wählen Sie Schließen, um das Dialogfeld mit den Transformationsregeln zu schließen.

Sie können das Umschaltssymbol verwenden, um eine Migrationsregel zu deaktivieren, ohne sie zu löschen. Sie können das Kopiersymbol verwenden, um eine bestehende Migrationsregel zu duplizieren. Sie können das Stiftsymbol verwenden, um eine bestehende Migrationsregel zu bearbeiten. Sie können das Löschsymbolsymbol verwenden, um eine bestehende Migrationsregel zu

löschen. Um alle Änderungen zu speichern, die Sie an Ihren Migrationsregeln vornehmen, wählen Sie **Alle speichern**.

Migrationsregeln exportieren

Wenn Sie AWS Database Migration Service (AWS DMS) verwenden, um Ihre Daten von Ihrer Quelldatenbank in Ihre Zieldatenbank zu migrieren, können Sie Informationen zu Ihren Migrationsregeln angeben AWS DMS. Weitere Informationen zu Aufgaben finden Sie unter [Arbeiten mit AWS Database Migration Service Replikationsaufgaben](#).

Um Migrationsregeln zu exportieren

1. Wählen Sie in der AWS Schema Conversion Tool im Menü Ansicht die Option Mapping-Ansicht.
2. Wählen Sie unter Migrationsregeln eine Migrationsregel aus und klicken Sie dann auf Migrationsregel ändern.
3. Wählen Sie Skript exportieren für AWS DMS.
4. Navigieren Sie zum gewünschten Speicherort des Skripts und klicken Sie auf Save. Ihre Migrationsregeln werden als JSON-Skript gespeichert, das von verwendet werden kann AWS DMS.

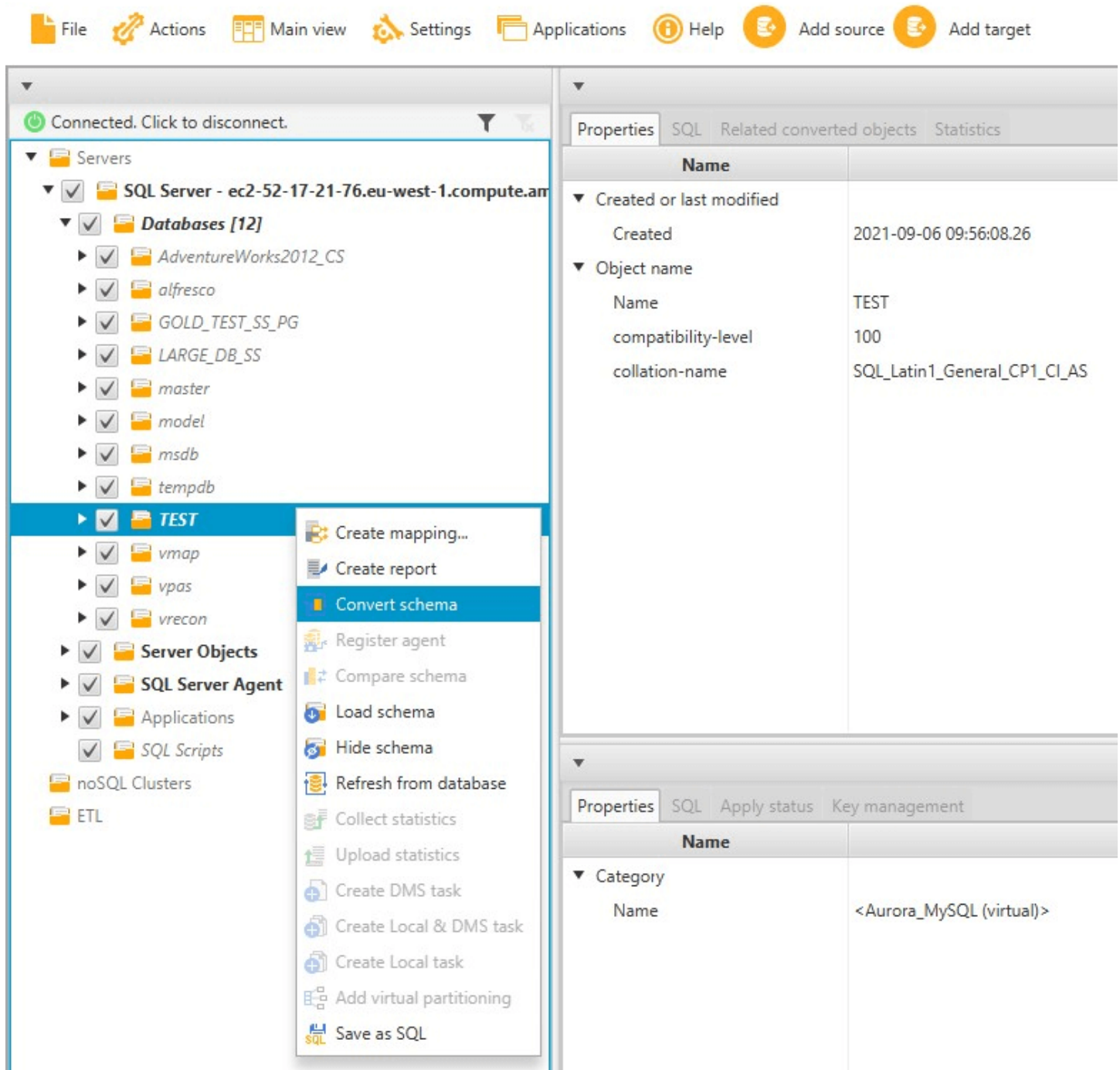
Konvertierung Ihres Schemas mit AWS SCT

Wenn Sie das Projekt sowohl mit der Quelldatenbank als auch mit der Zieldatenbank verknüpft haben, wird das Quelldatenbankschema im AWS Schema Conversion Tool-Projekt auf der linken Seite angezeigt. Das Schema wird in einer Strukturansicht dargestellt und jeder Knoten wird per Lazy Loading abgerufen. Wenn Sie einen Knoten in der Strukturansicht auswählen, fordert AWS SCT die Schemainformationen zu diesem Zeitpunkt aus der Quelldatenbank an.

Sie können Schemaelemente in der Quelldatenbank auswählen und dann das Schema in ein entsprechendes Schema für die Datenbank-Engine der Zieldatenbank konvertieren. Sie können beliebige Schemaelemente aus der Quelldatenbank für die Konvertierung auswählen. Falls das zu konvertierende Schemaelement von einem übergeordneten Element abhängig ist, generiert AWS SCT auch dafür ein entsprechendes Schema. Wenn beispielsweise eine Tabellenspalte konvertiert werden soll, generiert AWS SCT das Schema für die Spalte sowie für die Tabelle, in der sich die Spalte befindet, und für die Datenbank, in der sich die Tabelle befindet.

Konvertieren des Schemas

Um ein Schema aus Ihrer Quelldatenbank zu konvertieren, aktivieren Sie das Kontrollkästchen für den Namen des zu konvertierenden Schemas. Wählen Sie als Nächstes dieses Schema im linken Bereich Ihres Projekts aus. AWS SCT hebt den Schemanamen blau hervor. Öffnen Sie das Kontextmenü (Rechtsklick) für das Schema und wählen Sie Schema konvertieren, wie im Folgenden gezeigt.



Nachdem Sie das Schema von der Quelldatenbank konvertiert haben, können Sie aus dem linken Projektbereich Schemaelemente auswählen und das konvertierte Schema im mittleren Projektbereich anzeigen. Im mittleren Bereich werden unten die Eigenschaften des konvertierten Schemas und der SQL-Befehl zum Erstellen desselben wie folgt angezeigt:

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the project structure under 'Servers' > 'SQL Server - ec2-52-17-21-76.eu-west-1.cc' > 'Databases [12]' > 'LARGE_DB_SS' > 'Schemas [2]' > 'dbo' > 'Tables [4]' > 'Account'. The 'Account' table is selected. The main pane shows the SQL script for the source table:

```

1 CREATE TABLE [dbo].[Account] (
2 [ID] numeric(14,0) NOT NULL,
3 [AccountNo] varchar(16) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
4 [CurrencyID] numeric(3,0) NOT NULL,
5 [Description] varchar(160) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
6 [CustomerID] numeric(14,0) NOT NULL,
7 [StateID] numeric(2,0) NOT NULL,
8 [AccountBalance] numeric(14,3) NOT NULL,
9 [BlockedAmount] numeric(14,3) NOT NULL,
10 [Opendate] datetime NULL,
11 [Closedate] datetime NULL,
12 [RespManagerID] numeric(5,0) NULL,
13 [BankID] varchar(10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
14 )
15 ON [PRIMARY];

```

Below the script, it indicates 'Cursor position: 0 Modified: true'. The bottom pane shows the target Amazon RDS instance configuration for the MySQL table 'Account':

```

1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo VARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,

```

Nachdem das Schema konvertiert ist, können Sie das Projekt speichern. Die Schemainformationen von der Quelldatenbank werden zusammen mit dem Projekt gespeichert. Dank dieser Funktionalität können Sie offline ohne Verbindung zur Quelldatenbank arbeiten. AWS SCT stellt eine Verbindung zur Quelldatenbank her, um das Schema in Ihrem Projekt zu aktualisieren, wenn Sie Refresh from Database (Aktualisieren von Datenbank) für Ihre Quelldatenbank wählen. Weitere Informationen finden Sie unter [Aktualisierung und Aktualisierung Ihres konvertierten Schemas in AWS SCT](#).

Sie können einen Bewertungsbericht zur Datenbankmigration erstellen, in dem die nicht automatisch konvertierbaren Elemente aufgeführt werden. Mit diesem Bewertungsbericht können Sie die Schemaelemente ermitteln und auflösen, die nicht automatisch konvertiert werden können. Weitere Informationen finden Sie unter [Erstellung von Berichten zur Migrationsbewertung mit AWS SCT](#).

Wenn AWS SCT ein konvertiertes Schema generiert, wird dieses nicht sofort in die Zieldatenbank übernommen. Stattdessen wird das konvertierte Schema lokal gespeichert, bis Sie es für die

Zieldatenbank übernehmen. Weitere Informationen finden Sie unter [Wenden Sie Ihr konvertiertes Schema an](#).

Konvertiertes Schema bearbeiten

Sie können ein konvertiertes Schema bearbeiten und die Änderungen als Teil des Projekts speichern.

So bearbeiten Sie ein konvertiertes Schema

1. Wählen Sie im linken Projektbereich, in dem das Schema von der Quelldatenbank angezeigt wird, das Schemaelement aus, für das Sie das konvertierte Schema ändern möchten.
2. Wählen Sie unten im mittleren Bereich, in dem das konvertierte Schema für das ausgewählte Element angezeigt wird, die Registerkarte SQL aus.
3. Nehmen Sie im Text, der unter der Registerkarte SQL angezeigt wird, die erforderlichen Änderungen am Schema vor. Das Schema wird automatisch für das Projekt gespeichert, wenn Sie es aktualisieren.

```
▼ Target Amazon RDS for MySQL table: Account
Properties SQL Columns Partition columns Apply status Key management
1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo NVARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,
13 BankID VARCHAR(10) NOT NULL
14 );
```

Die Änderungen, die Sie am konvertierten Schema vorgenommen haben, werden für das Projekt gespeichert, wenn Sie es aktualisieren. Wenn Sie ein Schemaelement aus der Quelldatenbank neu konvertieren, aber das zuvor konvertierte Schema für dieses Element bereits aktualisiert hatten,

werden die erfolgten Aktualisierungen durch das auf Basis der Quelldatenbank neu konvertierte Schemaelement ersetzt.

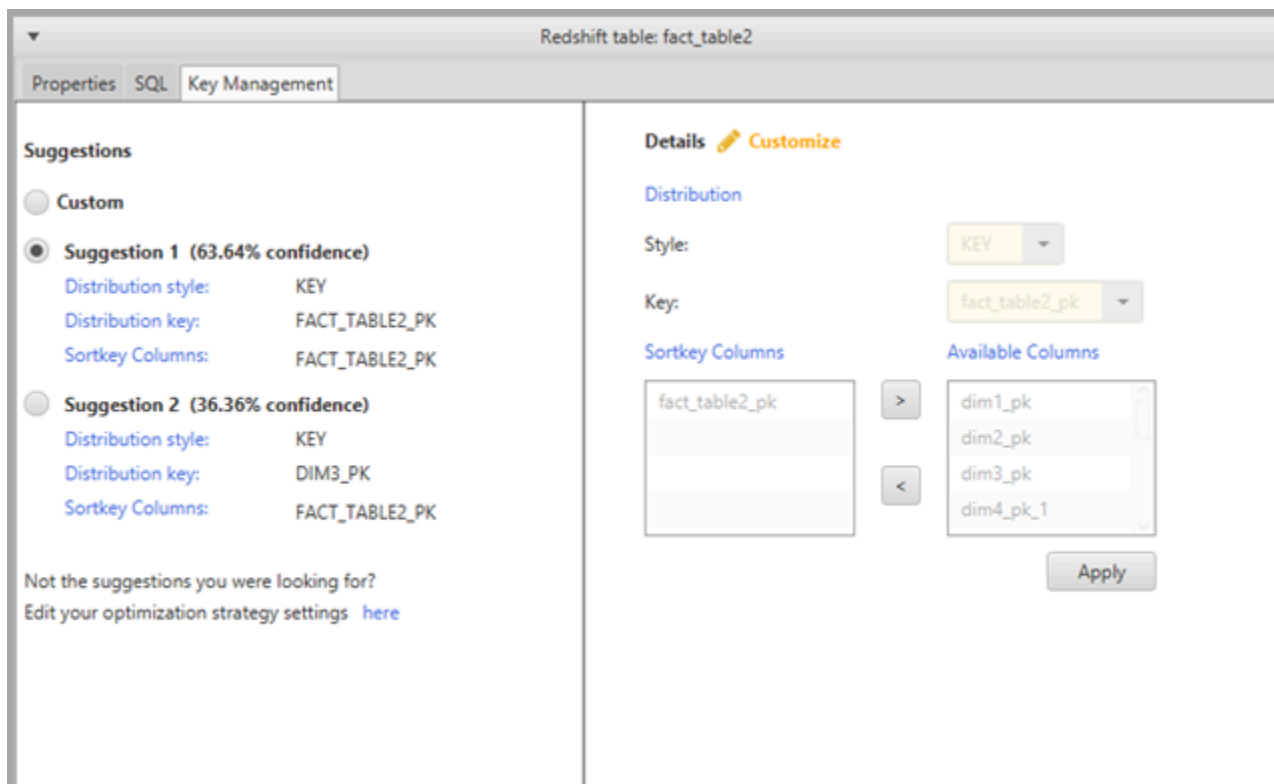
Löschen eines konvertierten Schemas

Bis Sie das Schema für die Zieldatenbank übernehmen, speichert AWS SCT das konvertierte Schema nur lokal in Ihrem Projekt. Sie können das geplante Schema aus dem Projekt löschen, indem Sie in der Strukturansicht auf den Knoten für die Zieldatenbank klicken und dann Refresh from Database auswählen. Da noch kein Schema auf die Zieldatenbank geschrieben wurde, führt die Aktualisierung von der Datenbank dazu, dass die geplanten Schemaelemente im AWS SCT-Projekt entfernt und durch das ersetzt werden, was bereits in der Zieldatenbank vorhanden ist.

Verwaltung und Anpassung von Schlüsseln in AWS SCT

Nachdem Sie das Schema mit dem AWS Schema Conversion Tool konvertiert haben, können Sie Ihre Schlüssel verwalten und bearbeiten. Die Schlüsselverwaltung spielt eine zentrale Rolle bei der Data Warehouse-Konvertierung.

Wählen Sie zum Verwalten von Schlüsseln eine Tabelle in Ihrer Zieldatenbank und dann die Registerkarte Key Management aus, wie im Folgenden gezeigt.



The screenshot displays the 'Key Management' tab for a Redshift table named 'fact_table2'. It is divided into two main panels:

- Suggestions:** This panel offers two suggestions for key management.
 - Suggestion 1 (63.64% confidence):** Distribution style: KEY, Distribution key: FACT_TABLE2_PK, Sortkey Columns: FACT_TABLE2_PK.
 - Suggestion 2 (36.36% confidence):** Distribution style: KEY, Distribution key: DIM3_PK, Sortkey Columns: FACT_TABLE2_PK. A link is provided to 'Edit your optimization strategy settings here'.
- Details:** This panel allows for customization of the key.
 - Distribution:** Style is set to 'KEY' and the Key is 'fact_table2_pk'.
 - Sortkey Columns:** A list box contains 'fact_table2_pk'. A right-pointing arrow (>) is next to it.
 - Available Columns:** A list box contains 'dim1_pk', 'dim2_pk', 'dim3_pk', and 'dim4_pk_1'. A left-pointing arrow (<) is next to it.
 - An 'Apply' button is located at the bottom right of the details section.

Der linke Bereich enthält Schlüsselvorschläge, einschließlich einer Zuverlässigkeitsbewertung der einzelnen Vorschläge. Sie können einen der Vorschläge auswählen oder einen Schlüssel anpassen, indem Sie diesen im rechten Bereich bearbeiten.

Entspricht die Auswahl nicht Ihren Erwartungen, können Sie Ihre Optimierungsstrategien bearbeiten und erneut eine Konvertierung ausprobieren. Weitere Informationen finden Sie unter [Auswahl von Optimierungsstrategien und Regeln für die Verwendung mit AWS SCT](#).

Verwandte Themen

- [Wählen Sie den besten Sortierschlüssel](#)
- [Wählen Sie den besten Vertriebsstil](#)

Erstellung und Verwendung des Bewertungsberichts in AWS SCT

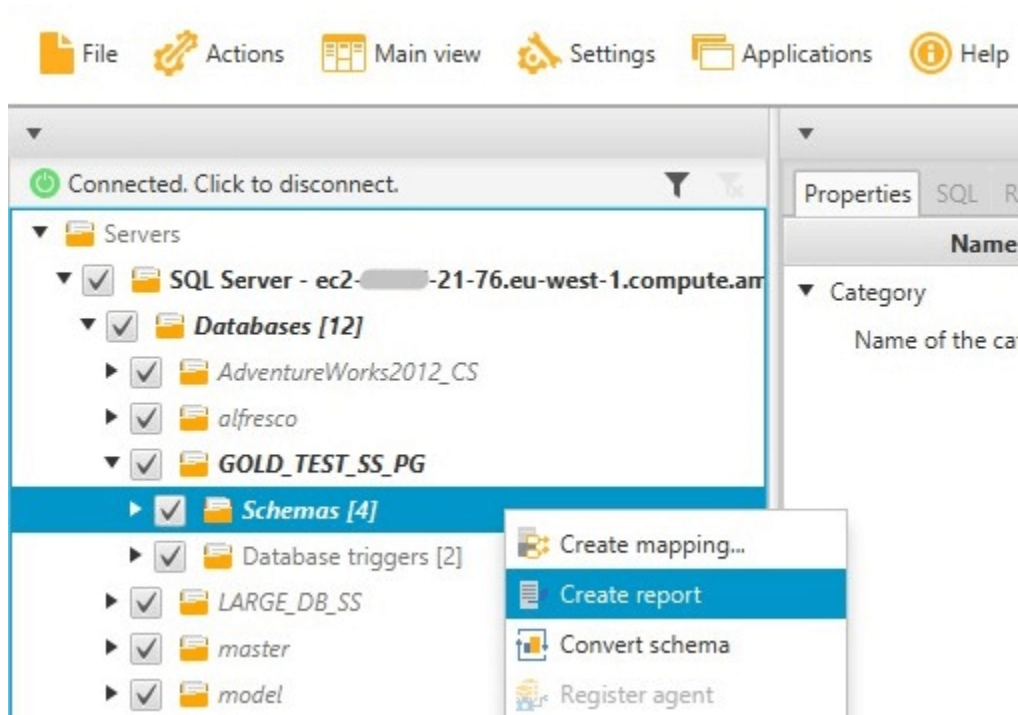
AWS Schema Conversion Tool erstellt einen Bericht zur Bewertung der Datenbankmigration, der Ihnen bei der Schemakonvertierung hilft. Dieser Bericht stellt wichtige Informationen zur Konvertierung des Schemas von Ihrer Quelldatenbank in die Zieldatenbank bereit. Der Bericht fasst alle Aufgaben der Schemakonvertierung zusammen und führt detailliert die Aktionselemente für Schemata auf, die nicht in die Engine der Zieldatenbank konvertiert werden können. Außerdem enthält der Bericht eine Aufwandsschätzung, da für die nicht automatisch konvertierbaren Elemente entsprechender Code in die Zieldatenbank geschrieben werden muss.

Erstellen eines Bewertungsberichts zur Datenbankmigration

Führen Sie die folgenden Schritte aus, um einen Bewertungsbericht zur Datenbankmigration zu erstellen.

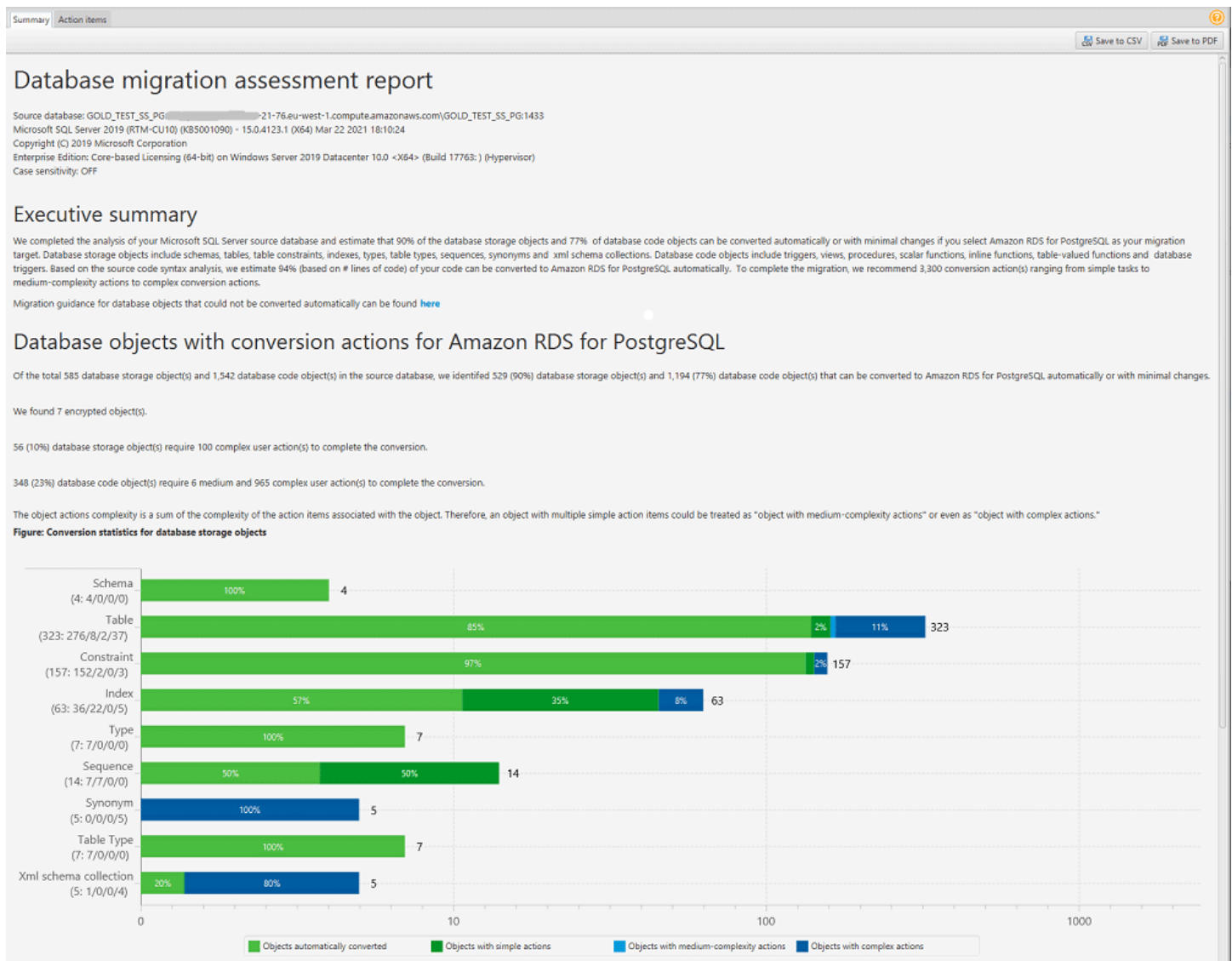
So erstellen Sie einen Bewertungsbericht zur Datenbankmigration

1. Wählen Sie im linken Bereich, in dem das Schema Ihrer Quelldatenbank angezeigt wird, ein Schemaobjekt aus, für das ein Bewertungsbericht erstellt werden soll.
2. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie Create Report aus.



Zusammenfassung des Bewertungsberichts

Nach der Erstellung des Bewertungsberichts wird die entsprechende Ansicht mit der Registerkarte Summary geöffnet. Auf der Registerkarte Summary wird eine Zusammenfassung des Berichts zur Bewertung der Datenbankmigration angezeigt. Es werden sowohl die automatisch konvertierten Elemente als auch Elemente, die nicht automatisch konvertiert wurden, angezeigt.



Für die Schemaelemente, die nicht automatisch in die Zieldatenbank-Engine konvertiert werden können, enthält die Zusammenfassung eine Aufwandsschätzung zur Erstellung von Schemaelementen in Ihrer Ziel-DB-Instance, die mit denen in der Quelle gleichgesetzt werden können.

Im Bericht wird der geschätzte Zeitaufwand für die Konvertierung dieser Schemaelemente wie folgt kategorisiert:

- Einfach — Aktionen, die in weniger als einer Stunde abgeschlossen werden können.
- Mittel — Aktionen, die komplexer sind und in ein bis vier Stunden abgeschlossen werden können.
- Signifikant — Aktionen, die sehr komplex sind und mehr als vier Stunden dauern, bis sie abgeschlossen sind.

Maßnahmenpunkte des Bewertungsberichts

Im Bewertungsbericht ist auch eine Registerkarte Action Items vorhanden. Diese Registerkarte enthält eine Liste der Elemente, die nicht automatisch in die Datenbank-Engine der Zieldatenbank konvertiert werden können. Wenn Sie ein Aktionselement aus der Liste auswählen, markiert AWS SCT das entsprechende Element im Schema, für das diese Aktion gilt.

Der Bericht enthält zudem Empfehlungen für die manuelle Konvertierung des Schemaelements. Weitere Informationen zum Festlegen der Handhabung von manuellen Konvertierungen finden Sie unter [Bearbeitung manueller Konvertierungen in AWS SCT](#).

The screenshot shows the AWS Schema Conversion Tool interface. The top menu includes File, Actions, Assessment Report view, Settings, Applications, Help, Add source, and Add target. The main window is divided into several sections:

- Summary / Action items:** A tabbed interface with 'Action items' selected.
- Left Sidebar:** A tree view showing the source database structure:
 - Servers: SQL Server - ec2-21-76.eu-west-1.compute.
 - Databases [12]: AdventureWorks2012_CS, alfresco, GOLD_TEST_SS_PG, LARGE_DB_SS, master, model, msdb, tempdb, TEST.
 - Schemas [1]: dbo.
 - Tables [8]: MSSQL_TemporalHistoryFor_1013578, MSSQL_TemporalHistoryFor_9655784, NonPartitionTable, PartitionTable, Position, test, tmp_tbl_sys_ver (System-Versioned), tmp_tbl_sys_ver_alter (System-Version).
 - Graph Tables, External Tables, Views.
 - Procedures [4]: P_INS_OUTP, POSITION_UPDATE_CASH_CGT_BULK, test2, test3.
 - SQL scalar functions, SQL table-valued functions, SQL inline functions.
- Right Panel:** A list of issues:
 - Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required. Recommended action: Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table. After the INSERT operation, you can make use of the rows saved in the temporary table. Number of occurrences: 1.
 - Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically. Recommended action: Use non-clustered indexes. Number of occurrences: 2.
 - Issue 794:** MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype. Recommended action: Please review generated code and modify it if necessary. Number of occurrences: 1.
 - Issue 826:** Check the default value for a DateTime variable. Recommended action: Check the default value for a DateTime variable. Number of occurrences: 1.
 - Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision. Recommended action: Review your transformed code and modify it if necessary to avoid a loss of accuracy. Number of occurrences: 8.
 - Issue 9997:** Unable to resolve objects. Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually. Number of occurrences: 3.
 - Issue 690:** MySQL doesn't support table types. Recommended action: Perform a manual conversion. Number of occurrences: 1.
 - Issue 811:** Unable to convert functions. Recommended action: Create a user-defined function. Number of occurrences: 12.
- Bottom Panel:** A detailed view of a source Microsoft SQL Server procedure:


```

1 create procedure POSITION_UPDATE_CASH_CGT_BULK
2   @InputPosNo tinyint readonly
3   , @posFlags bigint = 0
4   , @posFlagsMask bigint = 0
5 AS
6 update p
7 set   p.Flags = p.Flags & (~ @posFlagsMask ) | @posFlags
8 from   Position p
9       inner join @InputPosNo ipn on p.PosNo = ipn.F_POSNO
10
11 return 0
      
```
- Right Panel (Target):** Target Amazon RDS for MySQL category: Schemas. Properties: Name, Value.

Speichern des Bewertungsberichts

Sie können eine lokale Kopie des Berichts zur Bewertung der Datenmigration entweder als PDF-Datei oder als CSV-Datei speichern. Die CSV-Datei enthält nur Informationen zu den Aktionselementen. Die PDF-Datei enthält die Zusammenfassung und Informationen zu Aktionselementen, wie im folgenden Beispiel dargestellt.

Database objects with conversion actions for Amazon RDS for PostgreSQL

Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

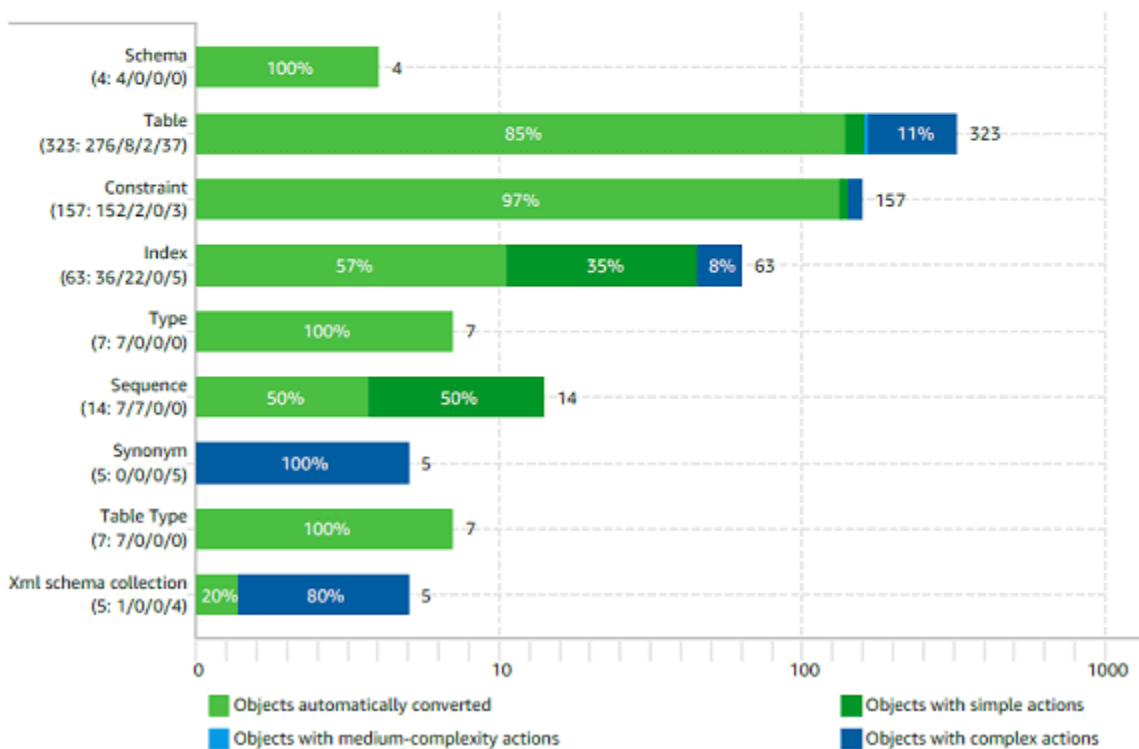
We found 7 encrypted object(s).

56 (10%) database storage object(s) require 100 complex user action(s) to complete the conversion.

348 (23%) database code object(s) require 6 medium and 965 complex user action(s) to complete the conversion.

The object actions complexity is a sum of the complexity of the action items associated with the object. Therefore, an object with multiple simple action items could be treated as "object with medium-complexity actions" or even as "object with complex actions."

Figure: Conversion statistics for database storage objects



Bearbeitung manueller Konvertierungen in AWS SCT

Der Bewertungsbericht enthält eine Liste der Elemente, die nicht automatisch in die Datenbank-Engine der Zieldatenbank konvertiert werden können. Für jedes nicht konvertierbare Element wird auf der Registerkarte Action Items ein Aktionselement angegeben.

Sie können wie folgt auf die Aktionselemente im Bewertungsbericht reagieren:

- Sie ändern das Schema der Quelldatenbank.

- Sie ändern das Schema der Zieldatenbank.

Ändern Ihres Quellschemas

Bei einigen Elementen ist es einfacher, das Datenbankschema der Quelldatenbank in ein Schema zu ändern, das automatisch konvertiert werden kann. Vergewissern Sie sich zunächst, dass die neuen Änderungen mit der Anwendungsarchitektur kompatibel sind, und aktualisieren Sie dann das Schema der Quelldatenbank. Aktualisieren Sie abschließend das Projekt mit den neuen Schemainformationen. Anschließend können Sie das aktualisierte Schema konvertieren und den Bewertungsbericht zur Datenbankmigration erneut erstellen. Für die im Quellschema geänderten Elemente werden keine Aktionselemente mehr angezeigt.

Der Vorteil dieses Verfahrens besteht darin, dass das aktualisierte Schema immer verfügbar ist, wenn Sie von der Quelldatenbank aktualisieren.

Ihr Zielschema ändern

Bei einigen Elementen ist es einfacher, das konvertierte Schema für die Zieldatenbank zu übernehmen und anschließend für die Elemente, die nicht automatisch konvertiert werden konnten, entsprechende Schemaelemente manuell zur Zieldatenbank hinzuzufügen. Durch Übernehmen des jeweiligen Schemas können Sie automatisch konvertierbare Schemata in die Zieldatenbank schreiben. Weitere Informationen finden Sie unter [Speichern und Anwenden Ihres konvertierten Schemas in AWS SCT](#).

Das Schema, das in die Zieldatenbank geschrieben wird, enthält nur automatisch konvertierbare Elemente. Nachdem Sie das Schema für die Zieldatenbank übernommen haben, können Sie manuell ein (mit der Quelldatenbank) gleichwertiges Schema in der Zieldatenbank erstellen. Die im Bewertungsbericht zur Datenbankmigration angegebenen Aktionselemente enthalten Vorschläge zum Erstellen eines gleichwertigen Schemas.

Warning

Wenn Sie ein Schema manuell in der Zieldatenbank erstellen, sollten Sie eine Kopie davon speichern. Falls Sie das konvertierte Schema vom Projekt erneut in die Zieldatenbank übernehmen, werden Ihre manuellen Änderungen überschrieben.

In einigen Fällen ist es nicht möglich, ein gleichwertiges Schema in der Zieldatenbank zu erstellen. Möglicherweise muss das Design der Anwendung und der Datenbank in Teilen überarbeitet werden, damit Sie die von der Engine bereitgestellte Funktionalität für Ihre Zieldatenbank nutzen können. In anderen Fällen kann das Schema, das nicht automatisch konvertiert werden kann, ignoriert werden.

Aktualisierung und Aktualisierung Ihres konvertierten Schemas in AWS SCT

Sie können sowohl das Quell- als auch das Zielschema im AWS Schema Conversion Tool-Projekt aktualisieren.

- **Quelle** — Wenn Sie das Schema für Ihre Quelldatenbank aktualisieren, AWS SCT ersetzen Sie das Schema in Ihrem Projekt durch das neueste Schema aus Ihrer Quelldatenbank. Mit dieser Funktion können Sie das Projekt aktualisieren, wenn das Schema der Quelldatenbank geändert wurde.
- **Ziel** — Wenn Sie das Schema für Ihre Zieldatenbank aktualisieren, wird das Schema in Ihrem Projekt durch das neueste Schema aus Ihrer Zieldatenbank AWS SCT ersetzt. Falls kein Schema für die Zieldatenbank übernommen wurde, löscht AWS SCT das konvertierte Schema aus dem Projekt. Anschließend können Sie das Schema von der Quelldatenbank für eine neue Zieldatenbank konvertieren.

Sie aktualisieren das Schema in Ihrem AWS SCT Projekt, indem Sie **Aus Datenbank aktualisieren** wählen.

Speichern und Anwenden Ihres konvertierten Schemas in AWS SCT

Wenn das AWS Schema Conversion Tool ein konvertiertes Schema generiert (wie in [Konvertierung Ihres Schemas mit AWS SCT](#) gezeigt), wird dieses nicht sofort für die Zieldatenbank übernommen. Stattdessen wird das konvertierte Schema lokal im Projekt gespeichert, bis Sie es für die Zieldatenbank übernehmen. Mit dieser Funktion können Sie mit Schemaelementen arbeiten, die sich nicht automatisch in die Zieldatenbank-Engine konvertieren lassen. Weitere Informationen zu Elementen, die nicht automatisch konvertierbar sind, finden Sie unter [Erstellung von Berichten zur Migrationsbewertung mit AWS SCT](#).

Optional können Sie das konvertierte Schema mithilfe des Tools als SQL-Skript in einer Datei speichern, bevor Sie das Schema für die Zieldatenbank übernehmen. Mit diesem Tool können Sie das konvertierte Schema auch direkt für die Zieldatenbank übernehmen.

Speichern Ihres konvertierten Schemas in einer Datei

Sie können das konvertierte Schema als SQL-Skripts in einer Textdatei speichern. Auf diese Weise können Sie die von AWS SCT generierten SQL-Skripts modifizieren, um solche Elemente zu berücksichtigen, die das Tool nicht automatisch konvertieren kann. Anschließend führen Sie die aktualisierten Skripts auf der DB-Ziel-Instance aus und übernehmen das konvertierte Schema in die Zieldatenbank.

Um Ihr konvertiertes Schema als SQL-Skript zu speichern

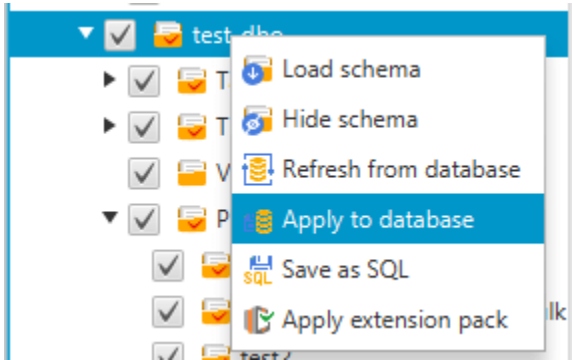
1. Wählen Sie Ihr Schema und öffnen Sie das Kontextmenü (Rechtsklick).
2. Wählen Sie Als SQL speichern.
3. Geben Sie den Namen der Datei ein und wählen Sie Speichern.
4. Speichern Sie Ihr konvertiertes Schema mit einer der folgenden Optionen:
 - Einzelne Datei
 - Eine Datei pro Stufe
 - Eine Datei pro Kontoauszug

Um das Format des SQL-Skripts auszuwählen

1. Wählen Sie im Menü Einstellungen die Option Projekteinstellungen.
2. Wählen Sie Scripts speichern.
3. Wählen Sie für Anbieter die Datenbankplattform aus.
4. Wählen Sie unter SQL-Skripts speichern in aus, wie Sie Ihr Datenbankschema-Skript speichern möchten.
5. Wählen Sie OK, um die Einstellungen zu speichern.

Wenden Sie Ihr konvertiertes Schema an

Um das konvertierte Schema für die Zieldatenbank zu übernehmen, wählen Sie im rechten Projektbereich das Schemaelement aus. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Schemaelement und wählen Sie Apply to database aus, wie hier gezeigt.



Das Schema des Erweiterungspakets

Wenn Sie zum ersten Mal ein konvertiertes Schema für die DB-Ziel-Instance übernehmen, fügt AWS SCT ein weiteres Schema zu Ihrer DB-Ziel-Instance hinzu. Über dieses Schema werden Systemfunktionen der Quelldatenbank implementiert, die zum Schreiben des konvertierten Schemas in die Ziel-DB-Instance benötigt werden. Das Schema wird als Erweiterungspaketschema bezeichnet.

Dieses Erweiterungspaketschema sollten Sie nicht ändern, andernfalls könnten im konvertierten Schema, das auf die DB-Ziel-Instance geschrieben wird, unerwünschte Ergebnisse auftreten. Nachdem das Schema vollständig zur DB-Ziel-Instance migriert wurde und Sie AWS SCT nicht mehr benötigen, können Sie das Erweiterungspaketschema löschen.

Das Erweiterungspaketschema wird wie folgt entsprechend Ihrer Quelldatenbank benannt:

- Greenplum: `aws_greenplum_ext`
- Microsoft SQL Server: `aws_sqlserver_ext`
- Netezza: `aws_netezza_ext`
- Oracle: `aws_oracle_ext`
- Snowflake: `aws_snowflake_ext`
- Teradata: `aws_teradata_ext`
- Vertica: `aws_vertica_ext`

Weitere Informationen finden Sie unter [Verwenden von AWS SCT Erweiterungspaketen](#).

Python-Bibliotheken

Um benutzerdefinierte Funktionen in Amazon Redshift zu erstellen, verwenden Sie die Sprache Python. Verwenden Sie das AWS SCT Erweiterungspaket, um Python-Bibliotheken für Ihre Amazon Redshift-Datenbank zu installieren. Weitere Informationen finden Sie unter [Verwenden von AWS SCT Erweiterungspaketen](#).

Optimierung von Amazon Redshift mithilfe von AWS SCT

Sie können das verwenden AWS Schema Conversion Tool, um Ihre Amazon Redshift-Datenbank zu optimieren. Wenn Sie Ihre Amazon Redshift-Datenbank als Quelle und eine Amazon Redshift-Testdatenbank als Ziel verwenden, AWS SCT empfiehlt es sich, Sortierschlüssel und Verteilungsschlüssel zu verwenden, um Ihre Datenbank zu optimieren.

Optimierung Ihrer Amazon Redshift-Datenbank

Gehen Sie wie folgt vor, um Ihre Amazon Redshift-Datenbank zu optimieren.

Um Ihre Amazon Redshift-Datenbank zu optimieren

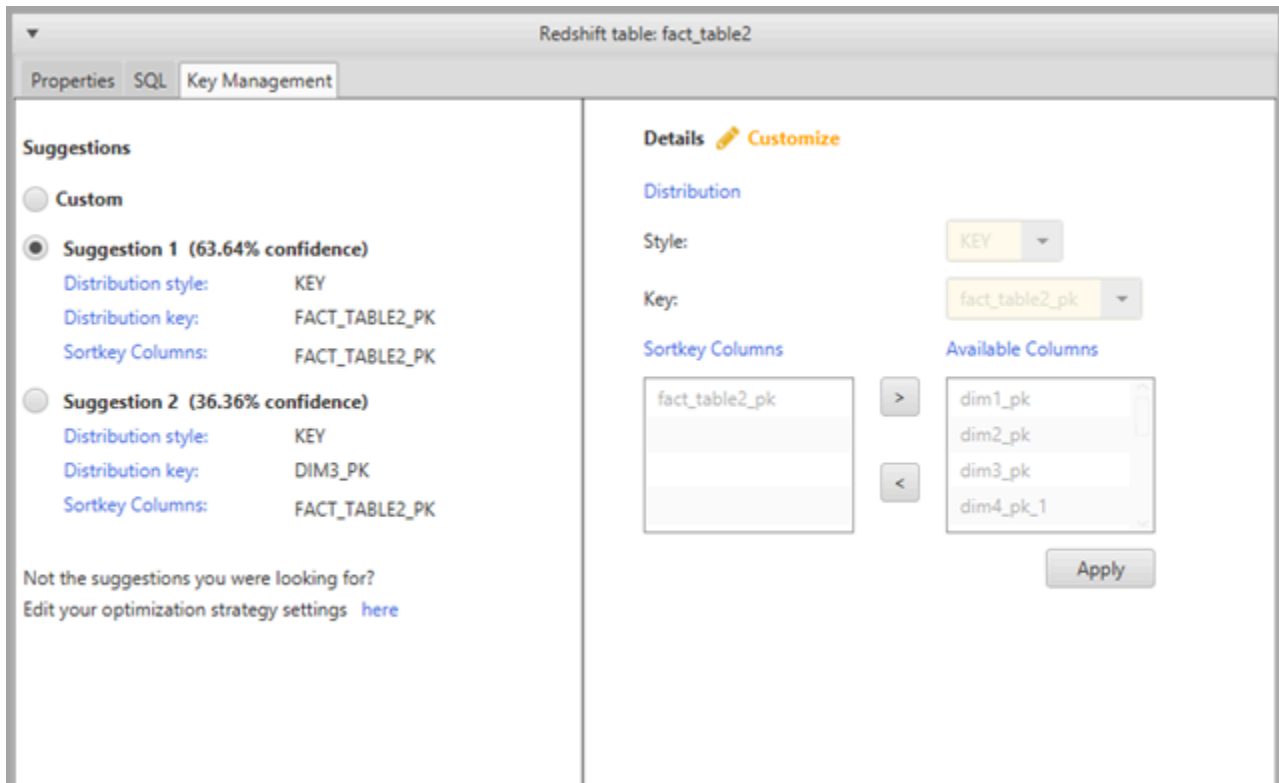
1. Erstellen Sie einen manuellen Snapshot Ihres Amazon Redshift-Clusters als Backup. Sie können den Snapshot löschen, nachdem Sie Ihren Amazon Redshift-Cluster optimiert und alle von Ihnen vorgenommenen Änderungen getestet haben. Weitere Informationen finden Sie unter [Amazon Redshift-Snapshots](#).
2. Wählen Sie im linken Bereich Ihres Projekts ein Schemaobjekt aus, von dem Sie konvertieren möchten. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie Collect Statistics aus.

AWS SCT verwendet die Statistiken, um Empfehlungen für Sortier- und Verteilungsschlüssel zu geben.

3. Wählen Sie im linken Bereich Ihres Projekts ein Schemaobjekt aus, von dem Sie optimieren möchten. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie Run Optimization aus.

AWS SCT gibt Empfehlungen für Sortier- und Verteilungsschlüssel.

4. Um diese Empfehlungen zu prüfen, erweitern Sie den Tabellenknoten unter dem Schema im linken Bereich des Projekts und wählen Sie eine Tabelle aus. Klicken Sie auf die Registerkarte Key Management, wie hier gezeigt.



Der linke Bereich enthält Schlüsselvorschläge, einschließlich einer Zuverlässigkeitsbewertung der einzelnen Vorschläge. Sie können einen der Vorschläge auswählen oder einen Schlüssel anpassen, indem Sie diesen im rechten Bereich bearbeiten.

5. Sie können einen Bericht mit den Optimierungsempfehlungen erstellen. Gehen Sie wie folgt vor, um den Bericht zu erstellen:

a. Wählen Sie im linken Bereich Ihres Projekts ein Schemaobjekt aus, von dem Sie optimiert haben. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie **Create Report** aus.

Der Bericht wird im Hauptfenster geöffnet und die Registerkarte **Summary** wird angezeigt. Die Anzahl der Objekte mit Optimierungsempfehlungen wird im Bericht angezeigt.

b. Klicken Sie auf die Registerkarte **Action Items**, um die wichtigsten Empfehlungen in einem Berichtsformat anzuzeigen.

c. Sie können eine lokale Kopie des Optimierungsberichts entweder als PDF-Datei oder als CSV-Datei speichern. Die CSV-Datei enthält nur Informationen zu den Aktionselementen. Die PDF-Datei enthält sowohl die Zusammenfassung als auch Informationen zu den Aktionselementen.

- Um die empfohlenen Optimierungen auf Ihre Datenbank anzuwenden, wählen Sie im rechten Bereich Ihres Projekts ein Objekt aus. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Objekt und wählen Sie Apply to database aus.

Konvertierung von Extrahieren, Transformieren und Laden (ETL) -Prozessen mitAWS Schema Conversion Tool

Sie können das verwendenAWS Schema Conversion Tool(AWS SCT), um ETL-Prozesse (Extrahieren, Transformieren und Laden) zu migrieren. Diese Art der Migration beinhaltet die Konvertierung von ETL-bezogener Geschäftslogik. Diese Logik kann sich entweder in Ihren Quell-Data Warehouses oder in externen Skripten befinden, die Sie separat ausführen.

DerzeitAWS SCTunterstützt die Konvertierung von ETL-Skripten in Objekte fürAWS Glueund Amazon Redshift RSQL, wie in der folgenden Tabelle dargestellt.

Quelle	Ziel
Informatica ETL-Skripts	Informatica
ETL-Pakete für Microsoft SQL Server Integration Services (SSIS)	AWS Glue oder AWS Glue Studio
Shell-Skripte mit eingebetteten Befehlen aus Teradata Basic Teradata Query (BTEQ)	Amazon-Redshift-RSQL
Teradata BTEQ ETL-Skripts	AWS Glueoder Amazon Redshift RSQL
TeradataFastExportJob-Skripte	Amazon-Redshift-RSQL
TeradataFastLoadJob-Skripte	Amazon-Redshift-RSQL
TeradataMultiLoadJob-Skripte	Amazon-Redshift-RSQL

Themen

- [Konvertierung von ETL-Prozessen inAWS Glue mitAWS SCT](#)
- [Konvertierung von ETL-Prozessen mithilfe der Python-API fürAWS Glue mitAWS SCT](#)
- [Konvertierung von Informatica ETL-Skripten mitAWS SCT](#)
- [SSIS konvertieren inAWS Glue mitAWS SCT](#)
- [SSIS konvertieren inAWS Glue Studio mitAWS SCT](#)

- [Konvertierung von Teradata BTEQ-Skripten zu Amazon Redshift RSQL mitAWS SCT](#)
- [Konvertierung von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen zu Amazon Redshift RSQL mitAWS SCT](#)
- [Teradata konvertierenFastExportJobskripte für Amazon Redshift RSQL mitAWS SCT](#)
- [Teradata konvertierenFastLoadJobskripte für Amazon Redshift RSQL mitAWS SCT](#)
- [Teradata konvertierenMultiLoadJobskripte für Amazon Redshift RSQL mitAWS SCT](#)

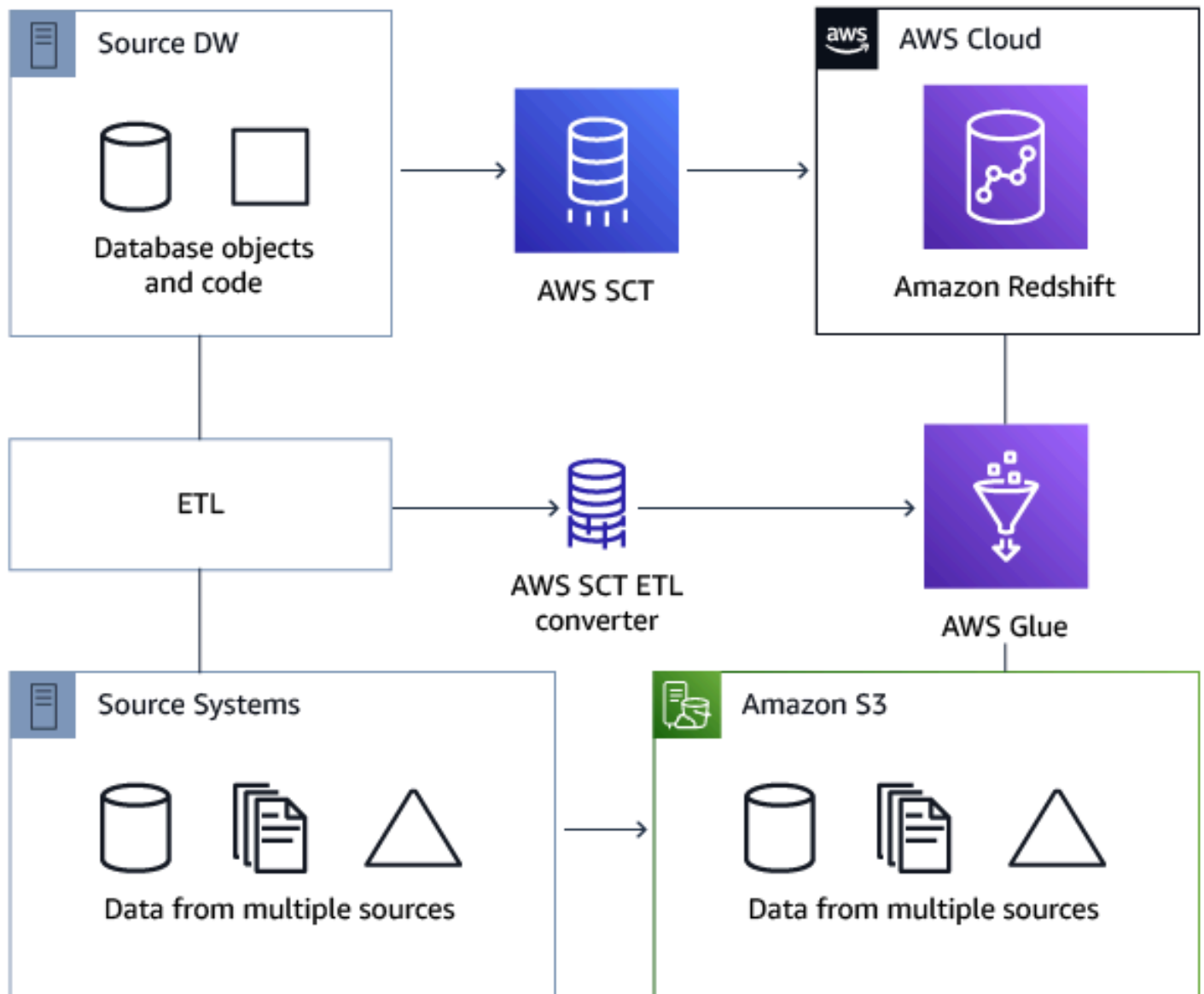
Konvertierung von ETL-Prozessen inAWS Glue mitAWS SCT

Im Folgenden finden Sie einen Überblick über den Prozess zum Konvertieren von ETL-Skripten inAWS Glue mitAWS SCT. In diesem Beispiel konvertieren wir eine Oracle-Datenbank zusammen mit den ETL-Prozessen, die für die Quelldatenbanken und Data Warehouses verwendet werden, in Amazon Redshift.

Themen

- [Voraussetzungen](#)
- [Grundlegendes zum AWS Glue-Datenkatalog](#)
- [Einschränkungen für die Konvertierung mitAWS SCT mitAWS Glue](#)
- [Schritt 1: Erstellen eines neuen -Projekts](#)
- [Schritt 2: Erstellen Sie eineAWS GlueAufgabe](#)

Das folgende Architekturdiagramm zeigt ein Beispiel für ein Datenbankmigrationsprojekt, das die Konvertierung von ETL-Skripten inAWS Glue.



Voraussetzungen

Bevor Sie beginnen, führen Sie die folgenden Schritte aus:

- Migrieren Sie alle Quelldatenbanken, zu denen Sie migrieren möchten AWS.
- Migrieren Sie die Ziel-Data Warehouses zu AWS.
- Erfassen Sie eine Liste des gesamten Codes Ihres ETL-Prozesses.
- Erstellen Sie eine Liste aller erforderlichen Verbindungsinformationen für jede Datenbank.

Außerdem AWS Glue benötigt Berechtigungen, um auf andere zuzugreifen AWS Ressourcen in Ihrem Namen. Diese Berechtigungen stellen Sie mithilfe von AWS Identity and Access Management (IAM) zur Verfügung. Stellen Sie sicher, dass Sie eine IAM-Richtlinie für erstellt haben AWS Glue. Weitere Informationen finden Sie unter [Erstellen Sie eine IAM-Richtlinie für AWS Kleberservice](#) in der AWS Glue Leitfaden für Entwickler.

Grundlegendes zum AWS Glue-Datenkatalog

Im Rahmen des Konvertierungsprozesses lädt AWS Glue Informationen zu den Quell- und Zieldatenbanken. Es organisiert diese Informationen in Kategorien in einer Struktur, die als Baum. Diese Struktur umfasst Folgendes:

- Verbindungen— Verbindungsparameter
- Crawler— Eine Liste von Crawlern, ein Crawler für jedes Schema
- Datenbanken— Behälter für Tische
- Tische— Metadatendefinitionen, die die Daten in den Tabellen darstellen
- Stellenangebote bei ETL— Geschäftslogik, die die ETL-Arbeit ausführt
- Auslöser— Logik, die steuert, wann ein ETL-Job ausgeführt wird AWS Glue (ob auf Anfrage, nach Zeitplan oder ausgelöst durch berufliche Ereignisse)

Der AWS Glue-Datenkatalog ist ein Index für die Speicherort-, Schema- und Laufzeitmetriken Ihrer Daten. Wenn Sie mit AWS Glue und AWS SCT arbeiten, enthält der AWS Glue-Datenkatalog Verweise auf Daten, die als Quelle und Ziel Ihrer ETL-Aufträge in AWS Glue verwendet werden. Um Ihr Data Warehouse zu erstellen, katalogisieren Sie diese Daten.

Sie verwenden die Informationen im Datenkatalog, um Ihre ETL-Aufträge zu erstellen und zu überwachen. Normalerweise führen Sie einen Crawler aus, um die Daten in Ihren Datenspeichern zu inventarisieren. Es gibt jedoch auch andere Möglichkeiten, Metadatentabellen in Ihren Datenkatalog einzufügen.

Wenn Sie eine Tabelle in Ihrem Datenkatalog definieren, fügen Sie sie einer Datenbank hinzu. Eine Datenbank wird verwendet, um Tabellen in AWS Glue zu organisieren.

Einschränkungen für die Konvertierung mit AWS SCT mit AWS Glue

Beim Konvertieren unter Verwendung von AWS SCT mit AWS Glue gelten die folgenden Einschränkungen:

Resource	Standardlimit
Anzahl der Datenbanken pro Konto	10.000
Anzahl der Tabellen pro Datenbank	100 000
Anzahl der Partitionen pro Tabelle	1 000 000
Anzahl der Tabellenversionen für jede Tabelle	100 000
Anzahl der Tabellen für jedes Konto	1 000 000
Anzahl der Partitionen für jedes Konto	10 000 000
Anzahl der Tabellenversionen für jedes Konto	1 000 000
Anzahl der Verbindungen für jedes Konto	1.000
Anzahl der Crawler für jedes Konto	25
Anzahl der Aufträge pro Konto	25
Anzahl der Auslöser für jedes Konto	25
Anzahl der gleichzeitigen Aufträge pro Konto	30
Anzahl der gleichzeitigen Auftragsausführungen für jeden Auftrag	3
Anzahl der Aufträge für jeden Auslöser	10
Anzahl der Entwicklungsendpunkte für jedes Konto	5
Maximale Anzahl an Datenverarbeitungseinheiten (DPUs), die von einem Entwicklungsendpunkt gleichzeitig genutzt werden	5
Maximale DPUs, die von einer Rolle gleichzeitig verwendet werden	100

Länge des Datenbanknamens	Unbegrenzt
	Aus Kompatibilitätsgründen mit anderen Metadaten Speichern, wie z. B. Apache Hive, wird der Name in Kleinbuchstaben gespeichert.
	Wenn Sie von Amazon Athena aus auf die Datenbank zugreifen möchten, geben Sie einen Namen an, der nur aus alphanumerischen Zeichen und Unterstrichen besteht.
Länge des Verbindungsnamens	Unbegrenzt
Länge des Crawler-Namens	Unbegrenzt

Schritt 1: Erstellen eines neuen -Projekts

Gehen Sie wie folgt vor, um ein neues Projekt zu erstellen:

1. Erstellen Sie ein neues Projekt in AWS SCT. Weitere Informationen finden Sie unter [Ein AWS SCT Projekt erstellen](#).
2. Fügen Sie Ihre Quell- und Zieldatenbanken zum Projekt hinzu. Weitere Informationen finden Sie unter [Hinzufügen von Datenbankservern zu einem AWS SCT Projekt](#).

Vergewissern Sie sich, dass Sie gewählt haben Benutzen AWS Glue in den Verbindungseinstellungen der Zieldatenbank. Wählen Sie dazu die AWS Glue Tab. Für Kopieren von AWS Profil, wählen Sie das Profil aus, das Sie verwenden möchten. Das Profil sollte automatisch das ausfüllen AWS Zugriffsschlüssel, geheimer Schlüssel und Amazon S3-Bucket-Ordner. Wenn dies nicht der Fall ist, geben Sie diese Informationen selbst ein. Nach Auswahl von OK analysiert AWS Glue die Objekte und lädt Metadaten in den AWS Glue-Datenkatalog.

Je nach Sicherheitseinstellungen erhalten Sie möglicherweise eine Warnmeldung, die Sie darüber informiert, dass Ihr Konto nicht über ausreichende Berechtigungen für einige der Schemata auf dem Server verfügt. Wenn Sie Zugriff auf die verwendeten Schemata haben, können Sie diese Meldung ignorieren.

3. Um die Vorbereitung des ETL-Imports abzuschließen, stellen Sie eine Verbindung zu Ihren Quell- und Zieldatenbanken her. Wählen Sie dazu Ihre Datenbank im Quell- oder Ziel-Metadatenbaum aus und wählen Sie dann Stellen Sie eine Verbindung zum Server her.

AWS Glue erstellt eine Datenbank auf dem Quelldatenbankserver und eine auf dem Zieldatenbankserver, um bei der ETL-Konvertierung zu helfen. Die Datenbank auf dem Zielsystem enthält den AWS Glue-Datenkatalog. Verwenden Sie die Suchfunktion im Quell- oder Zielbereich, um nach bestimmten Objekten zu suchen.

Um zu sehen, wie ein bestimmtes Objekt konvertiert wird, suchen Sie nach einem Objekt, das Sie konvertieren möchten, und wählen Sie Schema konvertieren aus seinem Kontextmenü (Rechtsklick). AWS SCT wandelt dieses ausgewählte Objekt in ein Skript um.

Sie können das konvertierte Skript auf der Seite überprüfen Skripte Ordner im rechten Bereich. Derzeit ist das Skript ein virtuelles Objekt, das nur als Teil Ihres AWS SCT-Projekts.

Um eine zu erstellen AWS Glue Job mit Ihrem konvertierten Skript, laden Sie Ihr Skript auf Amazon S3 hoch. Um das Skript auf Amazon S3 hochzuladen, wählen Sie das Skript aus und wählen Sie In S3 speichern aus seinem Kontextmenü (Rechtsklick).

Schritt 2: Erstellen Sie eine AWS Glue Aufgabe

Nachdem Sie das Skript in Amazon S3 gespeichert haben, können Sie es auswählen und dann konfigurieren AWS Glue über den Wizard zu öffnen, um das zu konfigurieren AWS Glue über den Wizard erleichtert die Einrichtung:

1. Auf der ersten Registerkarte des Wizards Datenfluss entwerfen, Sie können eine Ausführungsstrategie und die Liste der Skripte auswählen, die Sie in diesen einen Job aufnehmen möchten. Sie können für jedes Skript Parameter auswählen. Sie können die Skripts auch neu anordnen, damit sie in der richtigen Reihenfolge ausgeführt werden.
2. Auf der zweiten Registerkarte können Sie Ihrem Job einen Namen geben und die Einstellungen direkt konfigurieren für AWS Glue. Auf diesem Bildschirm können Sie die folgenden Einstellungen vornehmen:
 - AWS Identity and Access Management (IAM) -Rolle
 - Skript-Dateinamen und -Dateipfade
 - Verschlüsseln Sie das Skript mithilfe serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Schlüsseln (SSE-S3)

- Temporäres Verzeichnis
 - Generierter Python-Bibliothekspfad
 - Python-Bibliothekspfad des Benutzers
 - Pfad für die abhängigen .jar-Dateien
 - Pfad für referenzierte Dateien
 - Gleichzeitige DPU's für jede Auftragsausführung
 - Maximale Gleichzeitigkeit
 - Auftrag-Zeitüberschreitung (in Minuten)
 - Schwellenwert für die Verzögerungsbenachrichtigung (in Minuten)
 - Anzahl der Wiederholungen
 - Security configuration (Sicherheitskonfiguration)
 - Server-side encryption
3. Im dritten Schritt bzw. auf der dritten Registerkarte wählen Sie die konfigurierte Verbindung zum Zielpunkt aus.

Nachdem Sie den Auftrag fertig konfiguriert haben, wird er unter den ETL-Aufträgen im AWS Glue-Datenkatalog angezeigt. Wenn Sie den Auftrag auswählen, werden die zugehörigen Einstellungen angezeigt, sodass Sie sie überprüfen oder bearbeiten können. Um einen neuen Job zu erstellen in AWS Glue, wähle **Erstellen AWS Glue Beruf** aus dem Kontextmenü (Rechtsklick) für den Job. Dadurch wird die Schemadefinition angewendet. Um die Anzeige zu aktualisieren, wählen Sie **Refresh from database (Aus der Datenbank aktualisieren)** aus dem Kontextmenü.

An diesem Punkt können Sie den Auftrag in der AWS Glue-Konsole anzeigen. Melden Sie sich dazu bei der **AWS Management Console** und öffne die **AWS Glue Konsole** bei <https://console.aws.amazon.com/glue/>.

Sie können den neuen Auftrag testen, um sicherzustellen, dass er korrekt funktioniert. Überprüfen Sie die Daten in Ihrer Quelltable und prüfen Sie dann, ob die Zieltabelle leer ist. Führen Sie den Auftrag aus und überprüfen Sie erneut. Sie können die Fehlerprotokolle in der AWS Glue-Konsole einsehen.

Konvertierung von ETL-Prozessen mithilfe der Python-API für AWS Glue mit AWS SCT

In den folgenden Abschnitten finden Sie eine Beschreibung einer Konvertierung, bei der AWS Glue-API-Operationen in Python aufgerufen werden. Weitere Informationen finden Sie unter [Programm AWS Glue ETL-Skripte in Python](#) in der AWS Glue Leitfaden für Entwickler.

Themen

- [Schritt 1: Erstellen einer Datenbank](#)
- [Schritt 2: Verbindung erstellen](#)
- [Schritt 3: Erstellen Sie eine AWS Glue Crawler](#)

Schritt 1: Erstellen einer Datenbank

Der erste Schritt besteht darin, eine neue Datenbank in einer AWS Glue Datenkatalog mithilfe des [AWS SDK-API](#). Wenn Sie eine Tabelle im Datenkatalog definieren, fügen Sie sie einer Datenbank hinzu. Eine Datenbank wird verwendet, um Tabellen in AWS Glue zu organisieren.

Das folgende Beispiel zeigt die `create_database`-Methode der Python-API für AWS Glue:

```
response = client.create_database(  
    DatabaseInput={  
        'Name': 'database_name',  
        'Description': 'description',  
        'LocationUri': 'string',  
        'Parameters': {  
            'parameter-name': 'parameter value'  
        }  
    }  
)
```

Wenn Sie Amazon Redshift verwenden, wird der Datenbankname wie folgt gebildet.

```
{redshift_cluster_name}_{redshift_database_name}_{redshift_schema_name}
```

Der vollständige Name des Amazon Redshift-Clusters für dieses Beispiel lautet wie folgt.

```
rsddb03.apq1mpqso.us-west-2.redshift.amazonaws.com
```

Die folgende Abbildung zeigt ein Beispiel für einen gültigen Datenbanknamen. In diesem Fall ist `rsdbs03` der Name, bei dem es sich um den ersten Teil des vollständigen Namens des Cluster-Endpunkts handelt. Die Datenbank trägt den Namen `dev` und das Schema ist `ora_glue`.

```
rsdbs03_dev_ora_glue
```

Schritt 2: Verbindung erstellen

Erstellen Sie eine neue Verbindung in einem Datenkatalog mithilfe des [AWSSDK-API](#).

Das folgende Beispiel zeigt die Anwendung der [create_connection](#)-Methode der Python-API für AWS Glue:

```
response = client.create_connection(
    ConnectionInput={
        'Name': 'Redshift_abcde03.aabbcc112233.us-west-2.redshift.amazonaws.com_dev',
        'Description': 'Created from SCT',
        'ConnectionType': 'JDBC',
        'ConnectionProperties': {
            'JDBC_CONNECTION_URL': 'jdbc:redshift://aabbcc03.aabbcc112233.us-
west-2.redshift.amazonaws.com:5439/dev',
            'USERNAME': 'user_name',
            'PASSWORD': 'password'
        },
        'PhysicalConnectionRequirements': {
            'AvailabilityZone': 'us-west-2c',
            'SubnetId': 'subnet-a1b23c45',
            'SecurityGroupIdList': [
                'sg-000a2b3c', 'sg-1a230b4c', 'sg-aba12c3d', 'sg-1abb2345'
            ]
        }
    }
)
```

In `create_connection` werden die folgenden Parameter verwendet:

- `Name`(UTF-8-Zeichenfolge) — erforderlich. Für Amazon Redshift wird der Verbindungsname wie folgt gebildet: `Redshift_<Endpoint-name>_<redshift-database-name>`, zum Beispiel: `Redshift_abcde03_dev`
- `Description`(UTF-8-Zeichenfolge) — Ihre Beschreibung der Verbindung.

- `ConnectionType`(UTF-8-Zeichenfolge) — Erforderlich. Die Art der Verbindung. Derzeit wird nur JDBC unterstützt, SFTP wird nicht unterstützt.
- `ConnectionProperties`(dict) — Erforderlich. Eine Liste von Schlüssel-Wert-Paaren, die als Parameter für diese Verbindung verwendet werden, einschließlich der JDBC-Verbindungs-URL, des Benutzernamens und des Passworts.
- `PhysicalConnectionRequirements`(dict) — Physikalische Verbindungsanforderungen, die Folgendes beinhalten:
 - `SubnetId`(UTF-8-Zeichenfolge) — Die ID des von der Verbindung verwendeten Subnetzes.
 - `SecurityGroupIdList`(list) — Die von der Verbindung verwendete Sicherheitsgruppen-ID-Liste.
 - `AvailabilityZone`(UTF-8-Zeichenfolge) — Erforderlich. Die Availability Zone, die den Endpunkt enthält. Dieser Parameter ist veraltet.

Schritt 3: Erstellen Sie eine AWS GlueCrawler

Als Nächstes erstellen Sie einen AWS Glue-Crawler zum Auffüllen des AWS Glue-Katalogs. Weitere Informationen finden Sie unter [Tabellen mit einem Crawler katalogisieren](#) in der AWS Glue Leitfaden für Entwickler.

Der erste Schritt beim Hinzufügen eines Crawlers besteht darin, eine neue Datenbank in einem Datenkatalog zu erstellen, indem Sie den [AWS SDK-API](#). Bevor Sie beginnen, stellen Sie sicher, dass Sie zuerst alle früheren Versionen löschen, indem Sie den `delete_crawler` Betrieb.

Beim Erstellen des Crawlers gilt es einige Punkte zu berücksichtigen:

- Verwenden Sie für Crawler-Namen das Format `<redshift_node_name>_<redshift_database_name>_<redshift_schema_name>`, z. B.: `abcde03_dev_ora_glue`
- Verwenden Sie eine IAM-Rolle, die bereits vorhanden ist. Weitere Informationen zum Erstellen von IAM-Rollen finden Sie unter [IAM-Rollen erstellen](#) in der IAM-Benutzerhandbuch.
- Verwenden Sie den Namen der Datenbank, die Sie im vorherigen Schritt erstellt haben.
- Verwenden Sie den erforderlichen Parameter `ConnectionName`.
- Verwenden Sie für den Parameter `path` den Pfad des JDBC-Ziels, z. B.: `dev/ora_glue/%`

Im folgenden Beispiel wird ein vorhandener Crawler gelöscht und dann wird ein neuer Crawler mit der Python-API für AWS Glue erstellt:

```
response = client.delete_crawler(
    Name='crawler_name'
)

response = client.create_crawler(
    Name='crawler_name',
    Role='IAM_role',
    DatabaseName='database_name',
    Description='string',
    Targets={
        'S3Targets': [
            {
                'Path': 'string',
                'Exclusions': [
                    'string',
                ]
            },
        ],
        'JdbcTargets': [
            {
                'ConnectionName': 'ConnectionName',
                'Path': 'Include_path',
                'Exclusions': [
                    'string',
                ]
            },
        ],
    },
    Schedule='string',
    Classifiers=[
        'string',
    ],
    TablePrefix='string',
    SchemaChangePolicy={
        'UpdateBehavior': 'LOG'|'UPDATE_IN_DATABASE',
        'DeleteBehavior': 'LOG'|'DELETE_FROM_DATABASE'|'DEPRECATE_IN_DATABASE'
    },
    Configuration='string'
)
```

Nach der Crawler-Erstellung führen Sie einen Crawler aus, der eine Verbindung zu einem oder mehreren Datenspeichern herstellt, die Datenstrukturen bestimmt und Tabellen in den Datenkatalog schreibt. Sie können Ihren Crawler nach einem Zeitplan ausführen, wie im Folgenden gezeigt.

```
response = client.start_crawler(  
    Name='string'  
)
```

In diesem Beispiel wird Amazon Redshift als Ziel verwendet. Amazon Redshift-Datentypen sind zugeordnet AWS Glue Datentypen auf die folgende Weise, nachdem der Crawler ausgeführt wurde.

Amazon Redshift-Datentyp	AWS Glue -Datentyp
smallint	smallint
Ganzzahl	int
bigint	bigint
Dezimalwert	decimal(18,0)
decimal(p,s)	decimal(p,s)
real	double
double precision	double
Boolean	Boolean
char	string
varchar	string
varchar(n)	string
date	date
timestamp	timestamp
timestampz	timestamp

Konvertierung von Informatica ETL-Skripten mit AWS SCT

Du kannst benutzen AWS SCT Befehlszeilenschnittstelle (CLI) zur Konvertierung Ihrer Informatica ETL-Skripts, sodass Sie die Skripts mit Ihrer neuen Zieldatenbank verwenden können. Diese Konvertierung umfasst drei wichtige Schritte. Zuerst AWS SCT konvertiert den SQL-Code, der in Ihre Informatica-Objekte eingebettet ist. Als nächstes AWS SCT ändert die Namen von Datenbankobjekten gemäß den Migrationsregeln, die Sie in Ihrem Projekt angegeben haben. Schließlich AWS SCT leitet die Verbindungen Ihrer Informatica ETL-Skripts an die neue Zieldatenbank weiter.

Sie können Informatica ETL-Skripts als Teil von konvertieren AWS SCT Projekt zur Datenbankkonvertierung. Stellen Sie sicher, dass Sie Ihre Quell- und Zieldatenbanken zum Projekt hinzufügen, wenn Sie Informatica ETL-Skripts konvertieren.

Um Informatica ETL-Skripts zu konvertieren, stellen Sie sicher, dass Sie AWS SCT Version 1.0.667 oder höher. Machen Sie sich auch mit der Befehlszeilenschnittstelle von vertraut AWS SCT. Weitere Informationen finden Sie unter [AWS SCT CLI-Referenz](#).

Um Informatica ETL-Skripts zu konvertieren, verwenden Sie AWS SCT

1. Erstellen Sie ein neues AWS SCT CLI-Skript oder bearbeiten Sie eine vorhandene Szenariovorlage. Sie können zum Beispiel das herunterladen und bearbeiten `InformaticConversionTemplate.scts` Vorlage. Weitere Informationen finden Sie unter [CLI-Szenarien abrufen](#).
2. Laden Sie die erforderlichen JDBC-Treiber für Ihre Quell- und Zieldatenbanken herunter. Geben Sie den Standort dieser Treiber mithilfe des `SetGlobalSettings` Befehl. Geben Sie außerdem die Ordner an, in denen AWS SCT kann Protokolldateien speichern.

Das folgende Codebeispiel zeigt Ihnen, wie Sie den Pfad zu Oracle- und PostgreSQL-Treibern zum AWS SCT Einstellungen. Nachdem Sie dieses Codebeispiel ausgeführt haben, AWS SCT speichert Logfiles in der `C:\sct_log` Ordner. Außerdem AWS SCT speichert Konsolen-Logdateien in der `C:\Temp\oracle_postgresql` Ordner.

```
SetGlobalSettings
  -save: 'true'
  -settings: '{"oracle_driver_file": "C:\\drivers\\ojdbc8.jar",
    "postgresql_driver_file": "C:\\drivers\\postgresql-42.2.19.jar" }'
/

SetGlobalSettings
```

```
-save: 'false'
-settings: '{
"log_folder": "C:\\sct_log",
"console_log_folder": "C:\\Temp\\oracle_postgresql"}'
/
```

- Erstellen Sie ein neues AWS SCT-Projekt. Geben Sie den Namen und den Ort Ihres Projekts ein.

Das folgende Codebeispiel erstellt die `oracle_postgresql`-Projekt in der `C:\Temp`-Ordner.

```
CreateProject
-name: 'oracle_postgresql'
-directory: 'C:\Temp'
/
```

- Fügen Sie Verbindungsinformationen zu Ihren Quell- und Zieldatenbanken hinzu.

Das folgende Codebeispiel fügt Oracle- und PostgreSQL-Datenbanken als Quelle und Ziel für Ihre AWS SCT-Projekt.

```
AddSource
-password: 'source_password'
-port: '1521'
-vendor: 'ORACLE'
-name: 'ORACLE'
-host: 'source_address'
-database: 'ORCL'
-user: 'source_user'
/
AddTarget
-database: 'postgresql'
-password: 'target_password'
-port: '5432'
-vendor: 'POSTGRESQL'
-name: 'POSTGRESQL'
-host: 'target_address'
-user: 'target_user'
/
```

Ersetzen Sie im vorherigen Beispiel `quelle_benutzer` und `ziel_benutzer` mit den Namen Ihrer Datenbankbenutzer. Als nächstes ersetzen `quelle_passwort` und `Ziel-Passwort` mit

deinen Passwörtern. Für *quelladresse* und *ziel_adresse*, geben Sie die IP-Adressen Ihrer Quell- und Zieldatenbankserver ein.

Um eine Verbindung zu einer Oracle-Datenbank, Version 19 und höher, herzustellen, verwenden Sie den Oracle-Dienstnamen in der `AddSource` Befehl. Um dies zu tun, fügen Sie die `connectionTypeParameter` und setze seinen Wert auf `'basic_service_name'`. Fügen Sie dann das `serviceNameParameter` und setzen Sie seinen Wert auf Ihren Oracle-Servicenamen. Für weitere Informationen über die `AddSource` Befehl, siehe [AWS Schema Conversion Tool CLI-Referenz](#).

- Erstellen Sie ein neues AWS SCT Zuordnungsregel, die die Zieldatenbank-Engines für jedes Quelldatenbankschema definiert. Weitere Informationen finden Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#).

Im folgenden Codebeispiel wird eine Zuordnungsregel erstellt, die alle Oracle-Quelldatenbankschemas einschließt und PostgreSQL als Migrationsziel definiert.

```
AddServerMapping
  -sourceTreePath: 'Servers.ORACLE'
  -targetTreePath: 'Servers.POSTGRESQL'
/
```

- Fügen Sie Verbindungsinformationen zu Ihren XML-Quell- und Zieldateien von Informatica hinzu.

Das folgende Codebeispiel fügt die Informatica XML-Dateien aus dem `C:\Informatica_source` und `C:\Informatica_target` Ordner.

```
AddSource
  -name: 'INFA_SOURCE'
  -vendor: 'INFORMATICA'
  -mappingsFolder: 'C:\Informatica_source'
/
AddTarget
  -name: 'INFA_TARGET'
  -vendor: 'INFORMATICA'
  -mappingsFolder: 'C:\Informatica_target'
/
```

- Erstellen Sie eine weitere Zuordnungsregel, um die Informatica-XML-Zieldatei für Ihre Informatica-XML-Quelldatei zu definieren.

Im folgenden Codebeispiel wird eine Zuordnungsregel erstellt, die Quell- und Zieldateien von Informatica enthält, die im vorherigen Beispiel verwendet wurden.

```
AddServerMapping
-sourceTreePath: 'ETL.INFA_SOURCE'
-targetTreePath: 'ETL.INFA_TARGET'
/
```

8. Geben Sie die Datenbankserververbindung an, die der Informatica-Verbindungsnamenreferenz entspricht.

Im folgenden Codebeispiel wird die Umleitung Ihrer Informatica ETL-Skripts von Ihrer Quelle zur neuen Zieldatenbank konfiguriert. In diesem Beispiel werden auch Verbindungsvariablen konfiguriert.

```
ConfigureInformaticaConnectionsRedirect
-treePath: 'ETL.INFA_SOURCE.Files'
-connections: '{
"ConnectionNames": [
{
"name": "Oracle_src",
"newName": "postgres",
"treePath": "Servers.ORACLE"
}
]
"ConnectionVariables": [
{
"name": "$Source",
"treePath": "Servers.ORACLE"
}
]
}'
/
```

9. Konvertieren Sie Ihre Quelldatenbankschemas und Informatica ETL-Skripts.

Das folgende Codebeispiel konvertiert alle Ihre Oracle-Quelldatenbankschemas und Ihre Informatica XML-Datei.

```
Convert
-treePath: 'Servers.ORACLE.Schemas.%'
```

```
/
Convert
-treePath: 'ETL.INFA_SOURCE.Files'
/
```

10. (Optional) Speichern Sie Ihr Konvertierungsprojekt und den Bewertungsbericht. Dieser Bericht enthält die Konversionsmaßnahmen und Empfehlungen, wie die einzelnen Maßnahmen angegangen werden können.

Das folgende Codebeispiel speichert Ihr Projekt und speichert eine Kopie des Bewertungsberichts als PDF-Datei in der `C:\Temp` Ordner.

```
SaveProject
/
SaveReportPDF
-treePath: 'ETL.INFA_SOURCE.Files'
-file: 'C:\Temp\Informatica.pdf'
/
```

11. Speichern Sie Ihre konvertierte Informatica XML-Datei.

Das folgende Codebeispiel speichert die konvertierte XML-Datei in `C:\Temp` Ordner. Sie haben diesen Ordner im vorherigen Schritt mit dem `addTarget` Befehl.

```
SaveTargetInformaticaXML
-treePath: 'ETL.INFA_TARGET.Files'
/
```

12. Speichern Sie Ihr Skript als `.scts` Datei und starte es mit dem `runSCTBatch` Befehl in der AWS SCT CLI. Weitere Informationen finden Sie unter [AWS SCT CLI-Skriptmodus](#).

Im folgenden Beispiel wird der `Informatica.scts` Drehbuch in der `C:\Temp` Ordner. Sie können dieses Beispiel in Windows verwenden.

```
RunSCTBatch.cmd --pathtoscts "C:\Temp\Informatica.scts"
```

Wenn Sie Ihre Informatica-ETL-Quellskripte bearbeiten, führen Sie den AWS SCT Schon wieder CLI-Skript.

SSIS konvertieren in AWS Glue mit AWS SCT

Im Folgenden erfahren Sie, wie Sie Microsoft SQL Server Integration Services (SSIS) -Pakete konvertieren in AWS Glue unter Verwendung von AWS SCT.

Um Microsoft SSIS-Pakete zu konvertieren in AWS Glue, stellen Sie sicher, dass Sie AWS SCT Version 1.0.642 oder höher. Sie benötigen auch ein SSIS-Projekt mit ETL-Paketen — `.dtsx`, `.connmgr`, und `.params` Dateien im lokalen Ordner.

Sie benötigen keinen installierten SSIS-Server. Der Konvertierungsprozess erfolgt über die lokalen SSIS-Dateien.

Um ein SSIS-Paket zu konvertieren in AWS Glue unter Verwendung von AWS SCT

1. Erstellen Sie ein neues Projekt in AWS SCT oder öffnen Sie ein bestehendes Projekt. Weitere Informationen finden Sie unter [the section called “Erstellen eines Projekts”](#).
2. Wählen Sie **Quelle** hinzufügen aus dem Menü, um Ihrem Projekt ein neues SSIS-Quellpaket hinzuzufügen.
3. Wählen Sie **SQL Server-Integrationsdienst** und vervollständigen Sie das Folgende:
 - **Name der Verbindung**— Geben Sie den Namen für Ihre Verbindung ein. AWS SCT zeigt diesen Namen in der Metadatenstruktur an.
 - **Ordner für SSIS-Pakete**— Wählen Sie den Pfad zu Ihrem SSIS-Projektordner mit Paketen.

AWS SCT liest die Projektdateien (Dateien mit den Erweiterungen) `.dtsx`, `.connmgr` oder `.params`) aus dem lokalen Ordner und analysiert sie. Es organisiert sie dann zu einem AWS SCT Baum von Kategorien.

4. Wählen Sie **Ziel** hinzufügen aus dem Menü, um eine neue Zielplattform hinzuzufügen, um Ihre SSIS-Quellpakete zu konvertieren.
5. Wählen Sie **AWS Glue** und vervollständigen Sie das Folgende:
 - **Name der Verbindung**— Geben Sie den Namen für Ihre Verbindung ein. AWS SCT zeigt diesen Namen in der Metadatenstruktur an.
 - **Kopieren von AWS Profil**— Wählen Sie das zu verwendende Profil aus.
 - **AWS Zugriffstaste**— Geben Sie Ihre `AWS` Zugangsschlüssel.
 - **AWS geheimer Schlüssel**— Geben Sie Ihre `AWS` geheimer Schlüssel.
 - **Region**— Wählen Sie die `AWS`-Region die Sie aus der Liste verwenden möchten.

- Amazon S3-Bucket-Ordner— Geben Sie den Ordnerpfad für den Amazon S3-Bucket ein, den Sie verwenden möchten.

Sie können eine virtuelle verwenden AWS Glue Ziel. In diesem Fall müssen Sie die Verbindungsanmeldeinformationen nicht angeben. Weitere Informationen finden Sie unter [the section called "Virtuelle Ziele"](#).

6. Erstellen Sie eine neue Zuordnungsregel, die Ihr SSIS-Quellpaket und Ihr AWS Glue Ziel. Weitere Informationen finden Sie unter [the section called "Neue Regel"](#).
7. Auf der Ansehen Menü, wählen Hauptansicht.
8. Öffnen Sie in der SSIS-Baumansicht das Kontextmenü (Rechtsklick) für Verbindungsmanager, und wählen Sie dann Verbindungen konfigurieren.
9. Konfigurieren Sie den Projektverbindungsmanager.

Um eine Verbindungszuordnung für SSIS-Verbindungsmanager zu konfigurieren, geben Sie die AWS Glue Verbindung für den entsprechenden SSIS-Verbindungsmanager. Stellen Sie sicher, dass Ihr AWS Glue Verbindungen wurden bereits erstellt.

- a. Unter Verbindungen, wähle Verbindungen zum Projekt.
 - b. Für Kataloganschluss aufkleben, wähle das passende AWS Glue Verbindung.
10. Konfigurieren Sie den Paketverbindungsmanager:
 - a. Unter Verbindungen, wählen Sie Ihr Paket.
 - b. Für Kataloganschluss aufkleben, wähle das passende AWS Glue Verbindung.
 - c. Wiederholen Sie diese Aktionen für alle Verbindungen, die für Ihr Paket verfügbar sind.
 11. Wählen Sie Apply (Anwenden) aus.
 12. Konvertiere dein Paket. Suchen Sie in der Quellbaumansicht Pakete. Öffnen Sie das Kontextmenü (Rechtsklick) für Ihr Paket und wählen Sie Paket konvertieren.
 13. Speichern Sie das konvertierte Skript in Amazon S3. Suchen Sie in der Zielbaumansicht Skripte verpacken. Öffnen Sie das Kontextmenü (Rechtsklick) für Ihr konvertiertes Skript und wählen Sie In S3 speichern.
 14. Konfiguriere dein AWS Glue Beruf. Suchen Sie in der Zielbaumansicht Skripte verpacken. Öffnen Sie das Kontextmenü (Rechtsklick) für Ihr konvertiertes Skript und wählen Sie konfigurieren AWS Glue Aufgabe.
 15. Füllen Sie die drei Konfigurationsabschnitte aus:

a. Schließen Sie den Datenflussentwurfabschnitt:

- Strategie zur Umsetzung— Wählen Sie aus, wie Ihr Job ETL-Skripts ausführen soll. Wählen Sie SEQUENTIELLE um die Skripts in der Reihenfolge auszuführen, die im Assistenten angegeben ist. Wählen Sie PARALLEL um die Skripte parallel auszuführen, wobei die Reihenfolge, die im Wizard angegeben ist, ignoriert wird.
- Skripte— Wählen Sie den Namen Ihres konvertierten Skripts.
- Wählen Sie Weiter aus.

b. Schließen Sie den Eigenschaften des Arbeitsplatzesabschnitt:

- Name— Geben Sie den Namen Ihres AWS Glue Berufs.
- IAM-Rolle— Wählen Sie die IAM-Rolle aus, die für die Autorisierung von Ressourcen verwendet wird, die für die Ausführung des Jobs und den Zugriff auf Datenspeicher verwendet werden.
- Name der Skriptdatei— Geben Sie den Namen Ihres konvertierten Skripts ein.
- S3-Pfad der Skriptdatei— Geben Sie den Amazon S3-Pfad zu Ihrem konvertierten Skript ein.
- Verschlüsseln Sie das Skript mit SSE-S3— Wählen Sie diese Option, um Daten mithilfe serverseitiger Verschlüsselung mit von Amazon S3 verwalteten Verschlüsselungsschlüsseln (SSE-S3) zu schützen.
- Temporäres Verzeichnis— Geben Sie den Amazon S3-Pfad zu einem temporären Verzeichnis für Zwischenergebnisse ein. AWS Glue und AWS Glue Integrierte Transformationen verwenden dieses Verzeichnis, um in Amazon Redshift zu lesen oder zu schreiben.
- AWS S3 generiert automatisch den Pfad für Python-Bibliotheken. Sie können diesen Pfad überprüfen in Generierter Python-Bibliothekspfad. Sie können diesen automatisch generierten Pfad nicht bearbeiten. Um zusätzliche Python-Bibliotheken zu verwenden, geben Sie den Pfad ein Pfad der Python-Bibliothek des Benutzers.
- Pfad der Python-Bibliothek des Benutzers— Geben Sie die Pfade für zusätzliche Benutzer-Python-Bibliotheken ein. Trennen Sie Amazon S3-Pfade durch Kommas.
- Pfad für abhängige Jars— Geben Sie die Pfade für abhängige JAR-Dateien ein. Trennen Sie Amazon S3-Pfade durch Kommas.

- Pfad der referenzierten Dateien— Geben Sie die Pfade für zusätzliche Dateien ein, z. B. Konfigurationsdateien, die für Ihr Skript erforderlich sind. Trennen Sie Amazon S3-Pfade durch Kommas.
 - Maximale Kapazität— Geben Sie die maximale Anzahl von AWS Glue Datenverarbeitungseinheiten (DPU), die zugewiesen werden können, wenn dieser Job ausgeführt wird. Sie können eine Ganzzahl zwischen 2 und 100 eingeben. Der Standardwert ist 2.
 - Maximale Parallelität— Geben Sie die maximale Anzahl gleichzeitiger Läufe ein, die für diesen Job zulässig sind. Die Standardeinstellung ist 1. AWS Glue gibt einen Fehler zurück, wenn dieser Schwellenwert erreicht ist.
 - Zeitlimit für den Job (Minuten)— Geben Sie den Timeout-Wert für Ihren ETL-Job ein, um sich vor unerwarteten Jobs zu schützen. Der Standardwert für Batchaufträge beträgt 2880 Minuten (48 Stunden). Wenn der Job diesen Grenzwert überschreitet, ändert sich der Ausführungsstatus des Jobs in TIMEOUT.
 - Schwellenwert für verzögerte Benachrichtigungen (Minuten)— Geben Sie den Schwellenwert in Minuten zuvor ein, AWS Glue sendet eine Verzögerungsbenachrichtigung.
 - Anzahl der Wiederholungen— Geben Sie ein, wie oft (0–10) AWS Glue sollte den Job automatisch neu starten, falls er fehlschlägt. Jobs, die das Timeout-Limit erreichen, werden nicht neu gestartet. Der Standardwert ist 0.
 - Wählen Sie Weiter aus.
- c. Konfigurieren Sie die erforderlichen Verbindungen:
- i. Von Alle Verbindungen, wählen Sie die gewünschte AWS Glue Verbindungen und fügen sie der Liste von hinzugefügten Verbindungen hinzu.
 - ii. Wählen Sie Finish (Abschließen).
16. Erstellen Sie ein konfigurierbares AWS Glue Beruf. Suchen und erweitern Sie in der Zielbaumansicht ETL-Stellenangebote. Öffnen Sie das Kontextmenü (Rechtsklick) für den ETL-Job, den Sie konfiguriert haben, und wählen Sie dann Erstellen AWS Glue Beruf.
17. Führen Sie den AWS Glue Beruf:
- a. Öffnen Sie die AWS Glue-Konsole unter <https://console.aws.amazon.com/glue/>.
 - b. Wählen Sie im Navigationsbereich die Option Jobs (Aufträge) aus.
 - c. Wählen Sie Job hinzufügen, und wählen Sie dann den Job aus, den Sie ausführen möchten.
 - d. Auf der Aktionen-Tabulator, wählen Sie Job ausführen.

SSIS-Komponenten, die AWS SCT kann konvertieren in AWS Glue

Du kannst benutzen AWS SCT um Datenfluss- und Kontrollflusskomponenten sowie Container, Parameter und Variablen zu konvertieren.

Zu den unterstützten Datenflusskomponenten gehören die folgenden:

- ADO NET-Ziel
- ADO NET-Quelle
- Aggregate
- Cache-Transformation
- Transformation der Charakterkarte
- Bedingte Split-Transformation
- Spaltentransformation kopieren
- Transformation der Datenkonvertierung
- Abgeleitete Spaltentransformation
- Excel-Ziel
- Excel-Quelle
- Spaltentransformation exportieren
- Ziel für Flatfiles
- Flatfile-Quelle
- Fuzzy-Lookup-Transformation
- Spaltentransformation importieren
- Transformation nachschlagen
- Zusammenführen An Transformation teilnehmen
- Transformation zusammenführen
- Multicast-Transformation
- ODBC-Ziel
- ODBC-Quelle
- OLE DB-Befehlstransformation
- OLE DB-Ziel

- OLE DB-Quelle
- Prozentuale Stichprobentransformation
- Pivot-Transformation
- Ziel der Raw-Datei
- Rohdateiquelle
- RecordSetZielort
- Transformation der Zeilenanzahl
- Transformation der Zeilenstichprobe
- Transformation sortieren
- SQL Server-Ziel
- Vereinigung der gesamten Transformation
- Transformation aufheben
- XML-Quelle

Zu den unterstützten Control Flow-Komponenten gehören die folgenden:

- Masseneinfüfungsaufgabe
- Aufgabe „Paket ausführen“
- Aufgabe „SQL ausführen“
- Aufgabe „T-SQL-Anweisung ausführen“
- Ausdrucksaufgabe
- Dateisystem-Aufgabe
- Aufgabe „Bediener benachrichtigen“
- E-Mail-Aufgabe senden

Zu den unterstützten SSIS-Containern gehören die folgenden:

- Für Loop Container
- Für jeden Loop-Behälter
- Sequenzcontainer

SSIS konvertieren in AWS Glue Studio mit AWS SCT

Du kannst benutzen AWS SCT um Microsoft SQL Server Integration Services (SSIS) -Pakete zu konvertieren in AWS Glue Studio.

Ein SSIS-Paket enthält die notwendigen Komponenten wie den Verbindungsmanager, Aufgaben, Kontrollfluss, Datenfluss, Parameter, Event-Handler und Variablen, um eine bestimmte ETL-Aufgabe (Extrahieren, Transformieren und Laden) auszuführen. AWS SCT konvertiert SSIS-Pakete in ein Format, das kompatibel ist mit AWS Glue Studio. Nachdem Sie Ihre Quelldatenbank auf die migriert haben in AWS Cloud, Sie können diese konvertierten ausführen in AWS Glue Studio Jobs zur Ausführung von ETL-Aufgaben.

Um Microsoft SSIS-Pakete zu konvertieren in AWS Glue Studio, stellen Sie sicher, dass Sie AWS SCT Version 1.0.661 oder höher.

Themen

- [Voraussetzungen](#)
- [Hinzufügen von SSIS-Paketen zu Ihrem AWS SCT Projekt](#)
- [Konvertierung von SSIS-Paketen in AWS Glue Studio mit AWS SCT](#)
- [Erstellen AWS Glue Studio Jobs, die den konvertierten Code verwenden](#)
- [Erstellen eines Bewertungsberichts für ein SSIS-Paket mit AWS SCT](#)
- [SSIS-Komponenten, die AWS SCT kann konvertieren in AWS Glue Studio](#)

Voraussetzungen

In diesem Abschnitt erfahren Sie mehr über die erforderlichen Aufgaben für die Konvertierung von SSIS-Paketen in AWS Glue. Zu diesen Aufgaben gehört das Erstellen von erforderlichen AWS Ressourcen in Ihrem Konto.

Du kannst benutzen AWS Identity and Access Management (IAM), um Richtlinien und Rollen zu definieren, die für den Zugriff auf Ressourcen erforderlich sind in AWS Glue Studio verwendet. Weitere Informationen finden Sie unter [IAM-Berechtigungen für die AWS Glue Studio Benutzer](#).

Danach AWS SCT konvertiert Ihre Quellskripte in AWS Glue Studio, laden Sie die konvertierten Skripts in einen Amazon S3-Bucket hoch. Stellen Sie sicher, dass Sie diesen Amazon S3-Bucket erstellen und ihn in AWS Einstellungen des Serviceprofils. Weitere Hinweise zum Erstellen eines S3-

Buckets finden Sie unter [Erstellen Sie Ihren ersten S3-Bucket](#) in der Amazon Simple Storage Service-Benutzerhandbuch.

Um sicherzustellen, dass AWS Glue Studio eine Verbindung zu Ihrem Datenspeicher herstellen, einen benutzerdefinierten Connector und eine Verbindung erstellen. Speichern Sie die Datenbankanmeldeinformationen außerdem in AWS Secrets Manager.

Um einen benutzerdefinierten Connector zu erstellen

1. Laden Sie den JDBC-Treiber für Ihren Datenspeicher herunter. Für weitere Informationen zu JDBC-Treibern, die AWS SCTNutzungen, siehe [Die erforderlichen Datenbanktreiber werden heruntergeladen](#).
2. Laden Sie diese Treiberdatei in Ihren Amazon S3-Bucket hoch. Weitere Informationen finden Sie unter [Laden Sie ein Objekt in Ihren Bucket hoch](#) in der Amazon Simple Storage Service-Benutzerhandbuch.
3. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Glue Studio-Konsole unter <https://console.aws.amazon.com/gluestudio/>.
4. Wählen Sie Steckverbinder, dann wähle Benutzerdefinierten Konnektor erstellen.
5. Für Anschluss S3 URL, wähle S3 durchstöbern, und wählen Sie die JDBC-Treiberdatei aus, die Sie in Ihren Amazon S3-Bucket hochgeladen haben.
6. Geben Sie eine Beschreibung ein Name für Ihren Stecker. Geben Sie z. B. ein **SQLServer**.
7. Für Art des Steckverbinders, wähle JDBC.
8. Für Name der Klasse, geben Sie den Namen der Hauptklasse für Ihren JDBC-Treiber ein. Geben Sie für SQL Server ein **com.microsoft.sqlserver.jdbc.SQLServerDriver**.
9. Für JDBC-URL-Basis, geben Sie die JDBC-Basis-URL ein. Die Syntax der JDBC-Basis-URL hängt von Ihrer Quelldatenbank-Engine ab. Verwenden Sie für SQL Server das folgende Format: **jdbc:sqlserver://\$<host>:\$<port>;databaseName=\$<dbname>;user=\$<username>;password=\$<password>**.

Stellen Sie sicher, dass Sie ersetzen **<host>**, **<port>**, **<dbname>**, **<username>**, und **<password>** mit deinen Werten.

10. Für Trennzeichen für URL-Parameter, geben Sie das Semikolon ein (;).
11. Wählen Sie Create Connector (Connector erstellen) aus.

Um Datenbankmeldeinformationen zu speichern in AWS Secrets Manager

1. Loggen Sie sich ein bei AWS Management Console und öffne die AWS Secrets Manager Konsole bei <https://console.aws.amazon.com/secretsmanager/>.
2. Wählen Sie Store a new secret (Ein neues Secret speichern).
3. Führen Sie auf der Seite Choose secret type (Secret-Typ auswählen) die folgenden Schritte aus:
 - a. Für Geheimer Typ, wähle den Andere Art von Geheimnis.
 - b. Für Schlüssel/Wert-Paare, geben Sie die folgenden Schlüssel ein: **host, port, dbname, username**, und **password**.

Geben Sie als Nächstes Ihre Werte für diese Schlüssel ein.

4. Auf der Secret konfigurieren Seite, geben Sie eine Beschreibung ein Geheimer Name. Geben Sie z. B. ei **SQL_Server_secret**.
5. Wählen Sie Weiter aus. Dann, auf der Rotation konfigurieren Seite, wählen Weiter wieder.
6. Prüfen Sie auf der Seite Review (Prüfen) die Secret-Details und wählen Sie Store (Speichern).

So stellen Sie eine Verbindung für Ihren Connector her

1. Melden Sie sich bei der AWS Management Console an und öffnen Sie die AWS Glue Studio-Konsole unter <https://console.aws.amazon.com/gluestudio/>.
2. Wählen Sie den Connector aus, für den Sie eine Verbindung erstellen möchten, und wählen Sie dann Verbindung herstellen.
3. Auf der Verbindung herstellen Seite, geben Sie eine Beschreibung ein Name für deine Verbindung. Geben Sie z. B. ei **SQL-Server-connection**.
4. Für AWS Geheim, wähle das Geheimnis, das du in erstellt hast AWS Secrets Manager.
5. konfigurieren Netzwerk-Optionen, dann wähle Verbindung herstellen.

Jetzt können Sie eine erstellen AWS Glue Studio Job mit einem benutzerdefinierten Konnektor.

Weitere Informationen finden Sie unter [Erstellen AWS Glue Studio Jobs](#).

Hinzufügen von SSIS-Paketen zu Ihrem AWS SCT Projekt

Sie können mehrere SSIS-Pakete zu einem einzigen hinzufügen AWS SCT projekt.

Um ein SSIS-Paket zu Ihrem hinzuzufügenAWS SCTProjekt

1. Erstellen Sie ein neues Projekt inAWS SCToder öffnen Sie ein bestehendes Projekt. Weitere Informationen finden Sie unter [the section called “Erstellen eines Projekts”](#).
2. Wählen SieQuelle hinzuzufügenaus dem Menü, und wählen Sie dannSQL Server-Integrationsdienste.
3. FürName der Verbindung, geben Sie einen Namen für Ihre SSIS-Pakete ein.AWS SCTzeigt diesen Namen in der Baumstruktur im linken Bereich an.
4. FürOrdner für SSIS-Pakete, geben Sie den Pfad zu dem Ordner mit den SSIS-Quellpaketen ein.
5. Wählen SieZiel hinzuzufügenaus dem Menü, und wählen Sie dannAWS Glue Studio.

Um eine Verbindung herzustellenAWS Glue Studio,AWS SCTbenutzt deinAWSProfil. Weitere Informationen finden Sie unter [Speichern vonAWS Serviceprofilen inAWS SCT](#).

6. Erstellen Sie eine Zuordnungsregel, die Ihr SSIS-Quellpaket und IhrAWS Glue StudioZiel. Weitere Informationen finden Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#).
7. ErstellenAWS Glue StudioVerbindungen in derAWS Glue StudioKonsole. Weitere Informationen finden Sie unter[Verbindungen für Konnektoren erstellen](#).
8. Wählen SieVerbindungsmanagerÖffnen Sie im linken Baum das Kontextmenü (Rechtsklick) und wählen Sie dannVerbindungen konfigurieren.

AWS SCTzeigt dieVerbindungen konfigurierenFenster.

9. Wählen Sie für jede SSIS-Quellverbindung eineAWS Glue StudioVerbindung.

Konvertierung von SSIS-Paketen inAWS Glue StudiomitAWS SCT

Im Folgenden erfahren Sie, wie Sie SSIS-Pakete konvertieren inAWS Glue Studiounter Verwendung vonAWS SCT.

Um ein SSIS-Paket zu konvertierenAWS Glue Studio

1. Fügen Sie Ihr SSIS-Paket zu Ihrem hinzuAWS SCTprojekt. Weitere Informationen finden Sie unter [Hinzufügen von SSIS-Paketen zu IhremAWS SCTProjekt](#).
2. Erweitern Sie im linken Bereich denETLundSSISKnoten.
3. Wählen SiePakete, öffnen Sie das Kontextmenü (Rechtsklick) und wählen SiePaket konvertieren.

AWS SCT konvertiert Ihre ausgewählten SSIS-Pakete in JSON-Dateien. Diese JSON-Objekte stellen einen Knoten in einem gerichteten azyklischen Graphen (DAG) dar. Finden Sie Ihre konvertierten Dateien in der Paket DAGs-Knoten im rechten Baum.

4. Wählen Sie Paket DAGs, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie In Amazon S3 speichern.

Jetzt können Sie diese Skripte verwenden, um Jobs in der AWS Glue Studio.

Erstellen AWS Glue Studio Jobs, die den konvertierten Code verwenden

Nachdem Sie Ihre SSIS-Quellpakete konvertiert haben, können Sie die konvertierten JSON-Dateien verwenden, um AWS Glue Studio Jobs.

Um eine zu erstellen AWS Glue Studio Aufgabe

1. Wählen Sie Paket DAGs Öffnen Sie im rechten Baum das Kontextmenü (Rechtsklick) und wählen Sie dann konfigurieren AWS Glue Studio Aufgabe.
2. (Optional) Wenden Sie das Erweiterungspaket an, das SSIS-Funktionen emuliert AWS Glue Studio.
3. Der konfigurieren AWS Glue Studio Aufgabe Fenster öffnet sich.

Schließen Sie das ab Grundlegende Eigenschaften des Jobs Abschnitt:

- Name— Geben Sie einen Namen Ihres AWS Glue Studio Beruf.
- Name der Skriptdatei— Geben Sie einen Namen für Ihr Job-Skript ein.
- Jobparameter— Fügen Sie Parameter hinzu und geben Sie deren Werte ein.

Wählen Sie Weiter aus.

4. Schließen Sie das ab Erweiterte Jobeigenschaften Abschnitt:
 - IAM-Rolle— Wählen Sie die IAM-Rolle, die für die Autorisierung verwendet wird AWS Glue Studio und greifen Sie auf Datenspeicher zu.
 - S3-Pfad der Skriptdatei— Geben Sie den Amazon S3-Pfad zu Ihrem konvertierten Skript ein.
 - Temporäres Verzeichnis— Geben Sie den Amazon S3-Pfad zu einem temporären Verzeichnis für Zwischenergebnisse ein. AWS Glue Studio verwendet dieses Verzeichnis, um in Amazon Redshift zu lesen oder zu schreiben.

- AWS SCT generiert automatisch den Pfad für Python-Bibliotheken. Sie können diesen Pfad überprüfen in Generierter Python-Bibliothekspfad. Sie können diesen automatisch generierten Pfad nicht bearbeiten. Um zusätzliche Python-Bibliotheken zu verwenden, geben Sie den Pfad ein Pfad der Python-Bibliothek des Benutzers.
- Pfad der Python-Bibliothek des Benutzers— Geben Sie die Pfade für zusätzliche Benutzer-Python-Bibliotheken ein. Trennen Sie Amazon S3-Pfade durch Kommas.
- Pfad für abhängige Jars— Geben Sie die Pfade für abhängige ein *.jar-Dateien. Trennen Sie Amazon S3-Pfade durch Kommas.
- Pfad der referenzierten Dateien— Geben Sie die Pfade für zusätzliche Dateien ein, z. B. Konfigurationsdateien, die für Ihr Skript erforderlich sind. Trennen Sie Amazon S3-Pfade durch Kommas.
- Typ des Arbeiters— Wählen Sie G.1X oder G.2X.

Wenn du wählst G.1X Jeder Worker ist 1 DPU zugeordnet (4 vCPU, 16 GB Arbeitsspeicher und 64 GB Festplatte).

Wenn du wählst G.2X Jeder Worker ist 2 DPU (8 vCPU, 32 GB Arbeitsspeicher und 128 GB Festplatte) zugeordnet.

- Angeforderte Anzahl von Arbeitern— Geben Sie die Anzahl der Mitarbeiter ein, die bei der Ausführung des Jobs zugewiesen werden.
- Maximale Parallelität— Geben Sie die maximale Anzahl gleichzeitiger Läufe ein, die für diesen Job zulässig sind. Die Standardeinstellung ist 1. AWS Glue gibt einen Fehler zurück, wenn dieser Schwellenwert erreicht ist.
- Zeitlimit für den Job (Minuten)— Geben Sie den Timeout-Wert für Ihren ETL-Job ein, um sich vor unerwarteten Jobs zu schützen. Die Standardeinstellung ist 2.880 Minuten (48 Stunden) für Batch-Jobs. Wenn der Job diesen Grenzwert überschreitet, ändert sich der Ausführungsstatus des Jobs in TIMEOUT.
- Schwellenwert für verzögerte Benachrichtigungen (Minuten)— Geben Sie den Schwellenwert in Minuten zuvor ein. AWS SCT sendet eine Verzögerungsbenachrichtigung.
- Anzahl der Wiederholungen— Geben Sie ein, wie oft (0–10) AWS Glue sollte den Job automatisch neu starten, falls er fehlschlägt. Jobs, die das Timeout-Limit erreichen, werden nicht neu gestartet. Der Standardwert ist 0.

Wählen Sie Finish (Abschließen).

AWS SCT konfiguriert Ihre ausgewählten AWS Glue Studio Jobs.

5. Deine konfigurierten Jobs findest du unter Stellenangebote bei ETL im rechten Baum. Wählen Sie Ihren konfigurierten Job aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Erstellen AWS Glue Studio Aufgabe.
6. Wählen Sie Status beantragen und stellen Sie sicher, dass Statuswert für Ihren Job ist Erfolg.
7. Öffne die AWS Glue Studio Konsole, wähle Auffrischen, und wähle deinen Job. Wählen Sie anschließend Run (Ausführen) aus.

Erstellen eines Bewertungsberichts für ein SSIS-Paket mit AWS SCT

Der Bericht zur Bewertung der ETL-Migration bietet Informationen zur Konvertierung Ihrer SSIS-Pakete in ein Format, das kompatibel mit AWS Glue Studio ist. Der Bewertungsbericht enthält Aktionspunkte für die Komponenten Ihrer SSIS-Pakete. Diese Aktionspunkte zeigen, welche Komponenten AWS SCT nicht automatisch konvertieren kann.

Um einen ETL-Migrationsbewertungsbericht zu erstellen

1. Erweitern Sie die SSIS Knoten unter ETL in der linken Leiste.
2. Wählen Sie Pakete, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Bericht erstellen.
3. Sehen Sie sich die Zusammenfassung Tab. Hier, AWS SCT zeigt die Zusammenfassung der Informationen aus dem ETL-Migrationsbewertungsbericht an. Es enthält Konvertierungsergebnisse für alle Komponenten Ihrer SSIS-Pakete.
4. (Optional) Speichern Sie eine lokale Kopie des ETL-Migrationsbewertungsberichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):
 - Um den ETL-Migrationsbewertungsbericht als PDF-Datei zu speichern, wählen Sie Als PDF speichern oben rechts.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Konvertierung von Skripten.

- Um den ETL-Migrationsbewertungsbericht als CSV-Datei zu speichern, wählen Sie In CSV speichern oben rechts.

AWS SCT erstellt drei CSV-Dateien. Diese Dateien enthalten Aktionselemente, Handlungsempfehlungen und eine geschätzte Komplexität des manuellen Aufwands, der zur Konvertierung der Skripts erforderlich ist.

5. Wählen Sie die **AktionspunkteTab**. Diese Registerkarte enthält eine Liste von Elementen, für die eine manuelle Konvertierung erforderlich ist **AWS Glue Studio**. Wenn Sie ein Aktionselement aus der Liste auswählen, **AWS SCT** hebt das Element aus Ihrem **SSIS-Quellpaket** hervor, für das sich das Aktionselement bezieht.

SSIS-Komponenten, die **AWS SCT** kann konvertieren in **AWS Glue Studio**

Du kannst benutzen **AWS SCT** um **SSIS-Datenflusskomponenten** und **-Parameter** zu konvertieren **AWS Glue Studio**.

Zu den unterstützten **Datenflusskomponenten** gehören die folgenden:

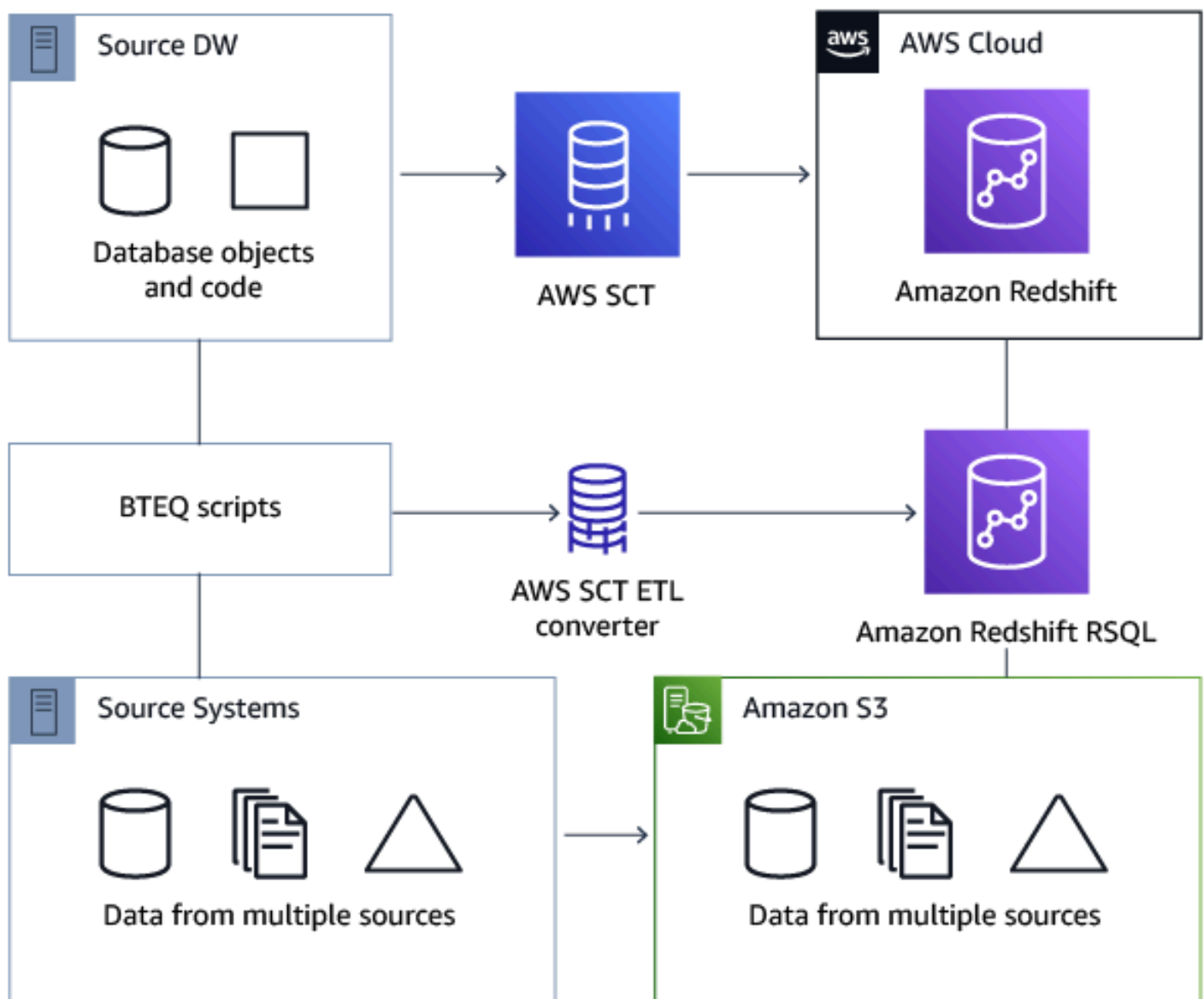
- ADO NET-Ziel
- ADO NET-Quelle
- Aggregate
- Charakterkarte
- Bedingte Aufteilung
- Spalte kopieren
- Datenkonvertierung
- Abgeleitete Spalte
- Nachschlagen
- Merge
- Merge Join
- Multicast
- ODBC-Ziel
- ODBC-Quelle
- OleDB-Ziel
- OLEDB-Quelle
- Anzahl der Zeilen
- Sortierung
- SQL Server-Ziel
- Alle vereinigen

AWS SCT kann mehr SSIS-Komponenten konvertieren in AWS Glue. Weitere Informationen finden Sie unter [SSIS-Komponenten, die AWS SCT kann konvertieren in AWS Glue](#).

Konvertierung von Teradata BTEQ-Skripten zu Amazon Redshift RSQL mit AWS SCT

Sie können das verwenden AWS Schema Conversion Tool (AWS SCT), um Teradata Basic Teradata Query (BTEQ) -Skripts in Amazon Redshift RSQL zu konvertieren.

Das folgende Architekturdiagramm zeigt das Datenbankmigrationsprojekt, das die Konvertierung von ETL-Skripten (Extract, Transform, Load) in Amazon Redshift RSQL beinhaltet.



Themen

- [Hinzufügen von BTEQ-Skripten zu Ihrem AWS SCT Projekt](#)
- [Konfiguration von Substitutionsvariablen in BTEQ-Skripten mit AWS SCT](#)
- [Konvertierung von Teradata BTEQ-Skripten zu Amazon Redshift RSQL mit AWS SCT](#)
- [Verwaltung von BTEQ-Skripten mit AWS SCT](#)
- [Erstellen eines Bewertungsberichts zur BTEQ-Skriptkonvertierung mit AWS SCT](#)
- [Bearbeiten und Speichern Ihrer konvertierten BTEQ-Skripte mit AWS SCT](#)

Hinzufügen von BTEQ-Skripten zu Ihrem AWS SCT Projekt

Sie können mehrere Skripte zu einem einzigen hinzufügen. AWS SCT Projekt.

Um ein BTEQ-Skript zu Ihrem hinzuzufügen AWS SCT Projekt

1. Erstellen Sie ein neues Projekt in AWS SCT oder öffnen Sie ein bestehendes Projekt. Weitere Informationen finden Sie unter [the section called “Erstellen eines Projekts”](#).
2. Wählen Sie Quelle hinzufügen aus dem Menü, und wählen Sie dann Teradata aus Ihre Quelldatenbank zum Projekt hinzuzufügen. Weitere Informationen finden Sie unter [Teradata als Quelle verwenden](#).
3. Wählen Sie Ziel hinzufügen über das Menü, um Ihrer Amazon Redshift-Zieldatenbank eine Amazon Redshift-Zieldatenbank hinzuzufügen AWS SCT Projekt.

Sie können eine virtuelle Amazon Redshift-Zieldatenbankplattform verwenden. Weitere Informationen finden Sie unter [Verwendung virtueller Ziele](#).

4. Erstellen Sie eine neue Zuordnungsregel, die Ihre Teradata-Quelldatenbank und Ihr Amazon Redshift-Ziel umfasst. Weitere Informationen finden Sie unter [Neue Zuordnungsregel hinzufügen](#).
5. Auf der Ansehen Menü, wählen Hauptansicht.
6. Erweitern Sie im linken Bereich den Skripte Knoten.
7. Wählen Sie BTEQ-Skripte, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Skripte laden.
8. Geben Sie den Speicherort des Quellcodes für Ihre Teradata BTEQ-Skripts ein und wählen Sie Ordner wählen.

AWS SCT zeigt die Skripte laden Fenster.

9. Führen Sie eine der folgenden Aktionen aus:

- a. Wenn Ihre Teradata BTEQ-Skripts die Substitutionsvariablen nicht enthalten, wählen Sie keine Substitutionsvariablen, und wählen Sie dann OKAY um Skripte zu Ihrem hinzuzufügen AWS SCT Projekt.
- b. Wenn Ihre Teradata BTEQ-Skripts die Substitutionsvariablen enthalten, konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter [Konfiguration von Substitutionsvariablen in BTEQ-Skripten](#).

Konfiguration von Substitutionsvariablen in BTEQ-Skripten mit AWS SCT

Ihre Teradata BTEQ-Skripts können Substitutionsvariablen enthalten. Sie können beispielsweise ein BTEQ-Skript mit Substitutionsvariablen verwenden, um denselben Befehlssatz in mehreren Datenbankumgebungen auszuführen. Du kannst benutzen AWS SCT um Substitutionsvariablen in Ihren BTEQ-Skripten zu konfigurieren.

Bevor Sie ein BTEQ-Skript mit Substitutionsvariablen ausführen, stellen Sie sicher, dass Sie die Werte für alle Variablen zuweisen. Dazu können Sie andere Tools oder Anwendungen wie ein Bash-Skript, UC4 (Automic) usw. verwenden. AWS SCT kann Substitutionsvariablen erst auflösen und konvertieren, nachdem Sie ihre Werte zugewiesen haben.

So konfigurieren Sie Substitutionsvariablen in Ihrem BTEQ-Skript

1. Fügen Sie Ihre BTEQ-Skripte zu Ihrem hinzu AWS SCT Projekt. Weitere Informationen finden Sie unter [Hinzufügen von BTEQ-Skripten zu Ihrem AWS SCT Projekt](#).

Wenn Sie Ihre Skripts hinzufügen, wählen Sie Substitutionsvariablen werden verwendet.

2. Für Variablenformat definieren, geben Sie einen regulären Ausdruck ein, der allen Substitutionsvariablen in Ihrem Skript entspricht.

Zum Beispiel, wenn die Namen Ihrer Substitutionsvariablen beginnen mit `{` und ende mit `}`, benutze den `\${\w+}` regulärer Ausdruck. Um Substitutionsvariablen abzugleichen, die entweder mit einem Dollarzeichen oder einem Prozentzeichen beginnen, verwenden Sie den `\$ | \% \w+` regulärer Ausdruck.

Reguläre Ausdrücke in AWS SCT entsprechen der Java-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie unter [java.util.regex Klassenmuster](#) in der Java-Dokumentation.

3. Wählen Sie OKAY um Skripte auf Ihre zu laden AWS SCT Projekt, und wählen Sie dann OKAY um das zu schließen Skripte laden Fenster.

4. Wählen Sie Variablenum alle entdeckten Substitutionsvariablen und ihre Werte einzusehen.
5. Für Wert, geben Sie den Wert für die Substitutionsvariable ein.

Konvertierung von Teradata BTEQ-Skripten zu Amazon Redshift RSQL mit AWS SCT

Im Folgenden erfahren Sie, wie Sie BTEQ-ETL-Skripts in Amazon Redshift RSQL konvertieren, indem Sie AWS SCT.

So konvertieren Sie ein Teradata BTEQ-Skript in Amazon Redshift RSQL

1. Fügen Sie Ihre BTEQ-Skripte zu Ihrem hinzu AWS SCT projekt. Weitere Informationen finden Sie unter [Hinzufügen von BTEQ-Skripten zu Ihrem AWS SCT Projekt](#).
2. Konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter [Konfiguration von Substitutionsvariablen in BTEQ-Skripten](#).
3. Erweitern Sie im linken Bereich den Skripte Knoten.
4. Führen Sie eine der folgenden Aktionen aus:
 - Um ein einzelnes BTEQ-Skript zu konvertieren, erweitern Sie den BTEQ-Skripte Knoten, wählen Sie das zu konvertierende Skript aus und wählen Sie dann Zu RSQL konvertieren aus dem Kontextmenü (Rechtsklick).
 - Um mehrere Skripts zu konvertieren, stellen Sie sicher, dass Sie alle zu konvertierenden Skripts auswählen. Dann wähle BTEQ-Skripte, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Zu RSQL konvertieren unter Skript konvertieren.

AWS SCT konvertiert alle Ihre ausgewählten Teradata BTEQ-Skripts in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Finden Sie Ihre konvertierten Skripte in der Skripte Knoten im Zieldatenbank-Panel.

5. Bearbeiten Sie Ihre konvertierten Amazon Redshift RSQL-Skripts oder speichern Sie sie. Weitere Informationen finden Sie unter [Bearbeiten und Speichern Ihrer konvertierten BTEQ-Skripte](#).

Verwaltung von BTEQ-Skripten mit AWS SCT

Sie können mehrere BTEQ-Skripts hinzufügen oder ein BTEQ-Skript aus Ihrem entfernen AWS SCT Projekt.

Um ein zusätzliches BTEQ-Skript zu Ihrem hinzuzufügen AWS SCT Projekt

1. Erweitern Sie die Skripte Knoten im linken Bereich.
2. Wählen Sie die BTEQ-Skripte Knoten, und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skripte laden.
4. Geben Sie die Informationen ein, die erforderlich sind, um ein neues BTEQ-Skript hinzuzufügen und Substitutionsvariablen zu konfigurieren. Weitere Informationen erhalten Sie unter [Hinzufügen von BTEQ-Skripten zu Ihrem AWS SCT Projekt](#) und [Konfiguration von Substitutionsvariablen in BTEQ-Skripten](#).

Um ein BTEQ-Skript aus Ihrem zu entfernen AWS SCT Projekt

1. Erweitern Sie die BTEQ-Skripte Knoten unter Skripte in der linken Leiste.
2. Wählen Sie das zu entfernende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skript löschen.

Erstellen eines Bewertungsberichts zur BTEQ-Skriptkonvertierung mit AWS SCT

Ein Bericht zur Bewertung der Konvertierung von BTEQ-Skripten enthält Informationen zur Konvertierung der BTEQ-Befehle und SQL-Anweisungen aus Ihren BTEQ-Skripten in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Der Bewertungsbericht enthält Aktionspunkte für BTEQ-Befehle und SQL-Anweisungen, die AWS SCT nicht konvertieren kann.

Um einen Bericht zur Bewertung der Konvertierung von BTEQ-Skripten zu erstellen

1. Erweitern Sie die BTEQ-Skripte Knoten unter Skripte in der linken Leiste.
2. Wählen Sie das zu konvertierende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Konvertierung zu RSQL unter Bericht erstellen.
4. Sehen Sie sich die Zusammenfassung Tab. Der Zusammenfassung Auf der Registerkarte werden die Informationen zur Zusammenfassung des BTEQ-Skripts aus dem Bewertungsbericht für

BTEQ-Skripts angezeigt. Es enthält Konvertierungsergebnisse für alle BTEQ-Befehle und SQL-Anweisungen aus Ihren BTEQ-Skripten.

5. (Optional) Speichern Sie eine lokale Kopie des Bewertungsberichts zur BTEQ-Skriptkonvertierung entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

- Um den Bericht zur Bewertung der BTEQ-Skriptkonvertierung als PDF-Datei zu speichern, wählen Sie **Als PDF speichern** oben rechts.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Konvertierung von Skripten.

- Um den Bericht zur Bewertung der BTEQ-Skriptkonvertierung als CSV-Datei zu speichern, wählen Sie **In CSV speichern** oben rechts.

Die CSV-Datei enthält Aktionselemente, empfohlene Aktionen und eine geschätzte Komplexität des manuellen Aufwands, der zur Konvertierung der Skripts erforderlich ist.

6. Wählen Sie die **Aktionspunkte** Tab. Diese Registerkarte enthält eine Liste von Elementen, die eine manuelle Konvertierung zu Amazon Redshift RSQL erfordern. Wenn Sie ein Aktionselement aus der Liste auswählen, hebt AWS SCT das Element aus Ihrem Quell-BTEQ-Skript hervor, für das sich das Aktionselement bezieht.

Bearbeiten und Speichern Ihrer konvertierten BTEQ-Skripte mit AWS SCT

Sie können Ihre konvertierten Skripte im unteren Bereich Ihres AWS SCT-Projekts. AWS SCT speichert das bearbeitete Skript als Teil Ihres Projekts.

Um Ihre konvertierten Skripte zu speichern

1. Erweitern Sie die **RSQL-Skripts** Knoten unter **Skripte** im **Zielfdatenbank**-Panel.
2. Wählen Sie Ihr konvertiertes Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie **Skript speichern**.
3. Geben Sie den Pfad zu dem Ordner ein, in dem das konvertierte Skript gespeichert werden soll, und wählen Sie **Speichern**.

AWS SCT speichert das konvertierte Skript in einer Datei und öffnet diese Datei.

Konvertierung von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen zu Amazon Redshift RSQL mitAWS SCT

Sie können das verwendenAWS Schema Conversion Tool(AWS SCT), um Shell-Skripts mit eingebetteten Teradata Basic Teradata Query (BTEQ) -Befehlen in Shell-Skripts mit eingebetteten Amazon Redshift RSQL-Befehlen zu konvertieren.

AWS SCTextrahiert Teradata BTEQ-Befehle aus Ihren Shell-Skripten und konvertiert sie in ein Format, das mit Amazon Redshift kompatibel ist. Nachdem Sie die Teradata-Datenbank zu Amazon Redshift migriert haben, können Sie diese konvertierten Skripts verwenden, um Ihre neue Amazon Redshift-Datenbank zu verwalten.

Sie können auch verwendenAWS SCTum Dateien mit Teradata BTEQ ETL-Skripten in Amazon Redshift RSQL zu konvertieren. Weitere Informationen finden Sie unter [Konvertierung von Teradata BTEQ-Skripten zu Amazon Redshift RSQL mitAWS SCT](#).

Themen

- [Hinzufügen von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen zu IhremAWS SCTProjekt](#)
- [Konfiguration von Substitutionsvariablen in Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mitAWS SCT](#)
- [Konvertierung von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mitAWS SCT](#)
- [Verwaltung von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mitAWS SCT](#)
- [Erstellen eines Bewertungsberichts für eine Shell-Skriptkonvertierung mitAWS SCT](#)
- [Bearbeiten und Speichern Ihrer konvertierten Shell-Skripte mitAWS SCT](#)

Hinzufügen von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen zu IhremAWS SCTProjekt

Sie können mehrere Skripte zu einem einzigen hinzufügenAWS SCTprojekt.

Um ein Shell-Skript zu Ihrem hinzuzufügenAWS SCTProjekt

1. Erstellen Sie ein neues Projekt inAWS SCToder öffnen Sie ein bestehendes Projekt. Weitere Informationen finden Sie unter [the section called “Erstellen eines Projekts”](#).

2. Wählen Sie **Quelle** hinzu~~f~~ügen aus dem Menü, und wählen Sie dann **Teradata** um Ihre Quelldatenbank zum Projekt hinzuzufügen. Weitere Informationen finden Sie unter [Teradata als Quelle verwenden](#).
3. Wählen Sie **Ziel** hinzu~~f~~ügen aus dem Menü und um eine Amazon Redshift-Zieldatenbank zu Ihrer AWS SCT projekt.

Sie können eine virtuelle Amazon Redshift-Zieldatenbankplattform verwenden. Weitere Informationen finden Sie unter [Verwendung virtueller Ziele](#).

4. Erstellen Sie eine neue Zuordnungsregel, die Ihre Teradata-Quelldatenbank und Ihr Amazon Redshift-Ziel umfasst. Weitere Informationen finden Sie unter [Neue Zuordnungsregel hinzufügen](#).
5. Auf der **Ansehen** Menü, wählen **Hauptansicht**.
6. Erweitern Sie im linken Bereich den **Skripte** Knoten.
7. Wählen Sie **Schale**, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie **Skripte laden**.
8. Geben Sie den Speicherort Ihrer Quell-Shell-Skripts mit eingebetteten Teradata BTEQ-Befehlen ein und wählen Sie **Ordner wählen**.

AWS SCT zeigt die **Skripte laden** Fenster.

9. Führen Sie eine der folgenden Aktionen aus:
 - Wenn Ihre Shell-Skripte die Substitutionsvariablen nicht enthalten, wählen Sie **Keine Substitutionsvariablen**, und wählen Sie dann **OKAY** um Skripte zu Ihrem hinzuzufügen AWS SCT projekt.
 - Wenn Ihre Shell-Skripte die Substitutionsvariablen enthalten, konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter [Konfiguration von Substitutionsvariablen in Shell-Skripten](#).

Konfiguration von Substitutionsvariablen in Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mit AWS SCT

Ihre Shell-Skripte können Substitutionsvariablen enthalten. Sie können beispielsweise ein einzelnes Skript mit Substitutionsvariablen verwenden, um Datenbanken in verschiedenen Umgebungen zu verwalten. Du kannst benutzen AWS SCT um Substitutionsvariablen in Ihren Shell-Skripten zu konfigurieren.

Bevor Sie BTEQ-Befehle mit Substitutionsvariablen aus einem Shell-Skript ausführen, stellen Sie sicher, dass Sie die Werte für alle Variablen in diesem Shell-Skript zuweisen. AWS SCT kann Substitutionsvariablen erst auflösen und konvertieren, nachdem Sie ihre Werte zugewiesen haben.

Um Substitutionsvariablen in Ihrem Shell-Skript zu konfigurieren

1. Fügen Sie Ihre Quell-Shell-Skripts zu Ihrem hinzu AWS SCT Projekt. Weitere Informationen finden Sie unter [Hinzufügen von Shell-Skripten zu Ihrem AWS SCT Projekt](#).

Wenn Sie Ihre Skripts hinzufügen, wählen Sie Substitutionsvariablen werden verwendet.

2. Für Variablenformat definieren, geben Sie einen regulären Ausdruck ein, der allen Substitutionsvariablen in Ihrem Skript entspricht.

Zum Beispiel, wenn die Namen Ihrer Substitutionsvariablen beginnen mit `{` und enden mit `}`, benutze den `\${\w+}` regulärer Ausdruck. Um Substitutionsvariablen abzugleichen, die entweder mit einem Dollarzeichen oder einem Prozentzeichen beginnen, verwenden Sie den `\$ \w+ | \% \w+` regulärer Ausdruck.

Reguläre Ausdrücke in AWS SCT entsprechen der Java-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie unter [java.util.regex Klassenmuster](#) in der Java-Dokumentation.

3. Wählen Sie OKAY um Skripte auf Ihre zu laden AWS SCT Projekt, und wählen Sie dann OKAY um das zu schließen Skripte laden Fenster.
4. Wählen Sie Variablen um alle entdeckten Substitutionsvariablen und ihre Werte einzusehen.
5. Für Wert, geben Sie den Wert für die Substitutionsvariable ein.

Konvertierung von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mit AWS SCT

Im Folgenden erfahren Sie, wie Sie Shell-Skripts mit eingebetteten Teradata BTEQ-Befehlen in Shell-Skripts mit eingebetteten Amazon Redshift RSQL-Befehlen konvertieren können, indem Sie AWS SCT.

Um ein Shell-Skript zu konvertieren

1. Fügen Sie Ihre Shell-Skripte zu Ihrem hinzu AWS SCT Projekt. Weitere Informationen finden Sie unter [Hinzufügen von Shell-Skripten zu Ihrem AWS SCT Projekt](#).

2. Konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter [Konfiguration von Substitutionsvariablen in Shell-Skripten](#).
3. Erweitern Sie im linken Bereich den SkripteKnoten.
4. Führen Sie eine der folgenden Aktionen aus:
 - Um BTEQ-Befehle aus einem einzelnen Shell-Skript zu konvertieren, erweitern Sie den SchaleKnoten, wählen Sie das zu konvertierende Skript aus und wählen Sie dann Skript konvertieren aus dem Kontextmenü (Rechtsklick).
 - Um mehrere Skripts zu konvertieren, stellen Sie sicher, dass Sie alle zu konvertierenden Skripts auswählen. Dann wähle Schale, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Skript konvertieren.
5. Wählen Sie OK.

AWS SCT konvertiert BTEQ-Befehle in Ihren ausgewählten Shell-Skripten in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Finden Sie Ihre konvertierten Skripte in der SkripteKnoten im Zieldatenbank-Panel.

6. Bearbeiten Sie Ihre konvertierten Amazon Redshift RSQL-Skripts oder speichern Sie sie. Weitere Informationen finden Sie unter [Bearbeiten und Speichern Ihrer konvertierten Shell-Skripte](#).

Verwaltung von Shell-Skripten mit eingebetteten Teradata BTEQ-Befehlen mit AWS SCT

Sie können mehrere Shell-Skripte hinzufügen oder ein Shell-Skript aus Ihrem AWS SCT projekt.

Um ein neues Shell-Skript zu Ihrem hinzuzufügen AWS SCT Projekt

1. Erweitern Sie die SkripteKnoten im linken Bereich.
2. Wählen Sie die SchaleKnoten, und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skripte laden.
4. Geben Sie die Informationen ein, die erforderlich sind, um ein neues Shell-Skript hinzuzufügen und Substitutionsvariablen zu konfigurieren. Weitere Informationen erhalten Sie unter [Hinzufügen von Shell-Skripten zu Ihrem AWS SCT Projekt](#) und [Konfiguration von Substitutionsvariablen in Shell-Skripten](#).

Um ein Shell-Skript aus Ihrem zu entfernenAWS SCTProjekt

1. Erweitern Sie dieSchaleKnoten unterSkriptein der linken Leiste.
2. Wählen Sie das zu entfernende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen SieSkript löschen.

Erstellen eines Bewertungsberichts für eine Shell-Skriptkonvertierung mitAWS SCT

DerBericht zur Bewertung der Konvertierung von Shell-Skriptenenthält Informationen zur Konvertierung der BTEQ-Befehle und SQL-Anweisungen. Die Konvertierung erfolgt von Ihren Quellskripten in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Der Bewertungsbericht enthält Aktionspunkte für BTEQ-Befehle und SQL-Anweisungen, dieAWS SCTkann nicht konvertieren.

Um einen Bericht zur Bewertung der Konvertierung von Shell-Skripten zu erstellen

1. Erweitern Sie dieSchaleKnoten unterSkripteim linken Bereich.
2. Wählen Sie das zu konvertierende Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dannBericht erstellen.
3. Sehen Sie sich dieZusammenfassungTab. DerZusammenfassungAuf der Registerkarte werden die zusammenfassenden Informationen aus dem Shell-Skript-Bewertungsbericht angezeigt. Es enthält Konvertierungsergebnisse für alle BTEQ-Befehle und SQL-Anweisungen aus Ihren Quellskripten.
4. (Optional) Speichern Sie eine lokale Kopie des Berichts zur Bewertung der Shell-Skript-Konvertierung entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

- Um den Bericht zur Bewertung der Shell-Skriptkonvertierung als PDF-Datei zu speichern, wählen SieAls PDF speichernoben rechts.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Konvertierung von Skripten.

- Um den Bericht zur Bewertung der Shell-Skriptkonvertierung als CSV-Datei zu speichern, wählen SieIn CSV speichernoben rechts.

Die CSV-Datei enthält Aktionselemente, empfohlene Aktionen und eine geschätzte Komplexität des manuellen Aufwands, der zur Konvertierung der Skripts erforderlich ist.

5. Wählen Sie die **AktionspunkteTab**. Diese Registerkarte enthält eine Liste von Elementen, die eine manuelle Konvertierung zu Amazon Redshift RSQL erfordern. Wenn Sie ein Aktionselement aus der Liste auswählen, hebt AWS SCT das Element aus Ihrem Quell-Shell-Skript hervor, für das sich das Aktionselement bezieht.

Bearbeiten und Speichern Ihrer konvertierten Shell-Skripte mit AWS SCT

Sie können Ihre konvertierten Skripte im unteren Bereich Ihres AWS SCT-Projekts. AWS SCT speichert das bearbeitete Skript als Teil Ihres Projekts.

Um Ihre konvertierten Skripte zu speichern

1. Erweitern Sie die **RSQL-SkriptsKnoten** unter **Skripte** im **Zieldatenbank-Panel**.
2. Wählen Sie Ihr konvertiertes Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie **Skript speichern**.
3. Geben Sie den Pfad zu dem Ordner ein, in dem das konvertierte Skript gespeichert werden soll, und wählen Sie **Speichern**.

AWS SCT speichert das konvertierte Skript in einer Datei und öffnet diese Datei.

Teradata konvertieren FastExportJobskripte für Amazon Redshift RSQL mit AWS SCT

Sie können das **AWS Schema Conversion Tool (AWS SCT)** um Teradata zu konvertieren **FastExportJobskripte** für Amazon Redshift RSQL.

Ein **FastExportJob-Skript** ist ein Satz von **FastExportBefehle** und **SQL-Anweisungen**, die Daten aus der Teradata-Datenbank auswählen und exportieren. AWS SCT konvertiert **FastExportBefehle** und **SQL-Anweisungen** in einem Format, das mit Amazon Redshift RSQL kompatibel ist. Nachdem Sie die Teradata-Datenbank zu Amazon Redshift migriert haben, können Sie diese konvertierten Skripts verwenden, um Daten aus der Amazon Redshift-Datenbank zu exportieren.

Themen

- [Hinzufügen FastExportJob-Skripte zu Ihrem AWS SCT Projekt](#)
- [Konfiguration von Substitutionsvariablen in Teradata FastExportJob-Skripte mit AWS SCT](#)
- [Teradata konvertieren FastExportJob-Skripte mit AWS SCT](#)

- [Verwaltung von TeradataFastExportJob-Skripte mitAWS SCT](#)
- [Erstellen eines Bewertungsberichts für einen TeradataFastExportJob-Script-Konvertierung mitAWS SCT](#)
- [Bearbeiten und Speichern Ihrer konvertierten TeradataFastExportJob-Skripte mitAWS SCT](#)

HinzufügenFastExportJob-Skripte zu IhremAWS SCTProjekt

Sie können mehrere Skripte zu einem einzigen hinzufügenAWS SCTprojekt.

Um eine hinzuzufügenFastExportJob-Skript zu IhremAWS SCTProjekt

1. Erstellen Sie ein neues Projekt inAWS SCToder öffnen Sie ein bestehendes Projekt. Weitere Informationen finden Sie unter [the section called “Erstellen eines Projekts”](#).
2. WähleQuelle hinzufügenaus dem Menü, und wählen Sie dannTeradataum Ihre Quelldatenbank zum Projekt hinzuzufügen. Weitere Informationen finden Sie unter [Teradata als Quelle verwenden](#).
3. WähleZiel hinzufügenaus dem Menü und um eine Amazon Redshift-Zieldatenbank zu IhrerAWS SCTprojekt.

Sie können eine virtuelle Amazon Redshift-Zieldatenbankplattform verwenden. Weitere Informationen finden Sie unter [Verwendung virtueller Ziele](#).

4. Erstellen Sie eine neue Zuordnungsregel, die Ihre Teradata-Quelldatenbank und Ihr Amazon Redshift-Ziel umfasst. Weitere Informationen finden Sie unter [Neue Zuordnungsregel hinzufügen](#).
5. Auf derAnsehenMenü, wählenHauptansicht.
6. Erweitern Sie im linken Bereich denSkripteKnoten.
7. WähleFastExport, öffnen Sie das Kontextmenü (Rechtsklick) und wählen SieSkripte laden.
8. Geben Sie den Speicherort des Quellcodes für Ihren Teradata einFastExportJob-Skripte und wähleOrdner wählen.

AWS SCTzeigt dieSkripte ladenFenster.

9. Führen Sie eine der folgenden Aktionen aus:
 - Wenn dein TeradataFastExportJobskripte enthalten die Substitutionsvariablen nicht, wählen SieKeine Substitutionsvariablenund dann wähleOKAYum Skripte zu Ihrem hinzuzufügenAWS SCTprojekt.

- Wenn dein TeradataFastExportJobskripte enthalten die Substitutionsvariablen, konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter [Konfiguration von Substitutionsvariablen inFastExportJob-Skripte](#).

Konfiguration von Substitutionsvariablen in TeradataFastExportJob-Skripte mitAWS SCT

Ihr TeradataFastExportJobskripte können Substitutionsvariablen enthalten. Sie können beispielsweise ein einzelnes Skript mit Substitutionsvariablen verwenden, um Daten aus mehreren Datenbanken zu exportieren. Du kannst benutzenAWS SCTum Substitutionsvariablen in Ihren Teradata-Skripten zu konfigurieren.

Bevor du einen läufstFastExportJob-Skript mit Substitutionsvariablen, stellen Sie sicher, dass Sie die Werte für alle Variablen zuweisen. Dazu können Sie andere Tools oder Anwendungen wie ein Bash-Skript, UC4 (Automic) usw. verwenden.AWS SCTkann Substitutionsvariablen erst auflösen und konvertieren, nachdem Sie ihre Werte zugewiesen haben.

Um Substitutionsvariablen in Ihrem zu konfigurierenFastExportJob-Skript

1. Fügen Sie Ihre Quelle hinzu TeradataFastExportJob-Skripte zu IhremAWS SCTprojekt. Weitere Informationen finden Sie unter [Hinzufügen von BTEQ-Skripten zu IhremAWS SCTProjekt](#).

Wenn Sie Ihre Skripts hinzufügen, wählen SieSubstitutionsvariablen werden verwendet.

2. FürVariablenformat definieren, geben Sie einen regulären Ausdruck ein, der allen Substitutionsvariablen in Ihrem Skript entspricht.

Zum Beispiel, wenn die Namen Ihrer Substitutionsvariablen beginnen mit\${und ende mit}, benutze den\\$\{\w+\}regulärer Ausdruck. Um Substitutionsvariablen abzugleichen, die entweder mit einem Dollarzeichen oder einem Prozentzeichen beginnen, verwenden Sie den\\$\w+|\%\w+regulärer Ausdruck.

Reguläre Ausdrücke in AWS SCT entsprechen der Java-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie unter[java.util.regex Klassenmuster](#)in der Java-Dokumentation.

3. WähleOKAYum Skripte auf Ihre zu ladenAWS SCTProjekt, und wählen Sie dannOKAYum das zu schließenSkripte ladenFenster.

4. Erweitern Sie im linken Bereich den SkripteKnoten. Wählen Sie FastExport, und wählen Sie dann Ihren Ordner mit Skripten aus. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Variablen exportieren unter Substitutionsvariablen.
5. Exportieren Sie Substitutionsvariablen für ein Skript. Erweitern Sie Ihren Ordner mit Skripten, wählen Sie Ihr Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Variablen exportieren unter Substitutionsvariablen.
6. Geben Sie den Namen der Datei mit kommagetrennten Werten (CSV) ein, um die Substitutionsvariablen zu speichern, und wählen Sie Speichern.
7. Öffnen Sie diese CSV-Datei und geben Sie die Werte für die Substitutionsvariablen ein.

Je nach Betriebssystem verwendet AWS SCT verschiedene Formate für CSV-Dateien. Die Werte in der Datei können entweder in Anführungszeichen stehen oder nicht. Stellen Sie sicher, dass Sie für die Werte der Substitutionsvariablen dasselbe Format verwenden wie für die anderen Werte in der Datei. AWS SCT kann die CSV-Datei mit Werten in verschiedenen Formaten nicht importieren.

8. Speichern Sie die CSV-Datei.
9. Erweitern Sie im linken Bereich den SkripteKnoten. Wählen Sie FastExport, und wählen Sie dann Ihr Skript aus. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Variablen importieren unter Substitutionsvariablen.
10. Wählen Sie Ihre CSV-Datei und wählen Sie dann Öffnen.
11. Wählen Sie Variablen um alle entdeckten Substitutionsvariablen und ihre Werte einzusehen.

Teradata konvertieren FastExportJob-Skripte mit AWS SCT

Im Folgenden erfahren Sie, wie Sie Teradata konvertieren FastExportJob zu Amazon Redshift RSQL mithilfe von AWS SCT.

Um einen Teradata zu konvertieren FastExportJobskript für Amazon Redshift RSQL

1. Fügen Sie Ihre FastExportJob-Skripte zu Ihrem AWS SCT-Projekt. Weitere Informationen finden Sie unter [Hinzufügen FastExportJob-Skripte zu Ihrem AWS SCT-Projekt](#).
2. Konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter [Konfiguration von Substitutionsvariablen in FastExportJob-Skripten](#).
3. Erweitern Sie im linken Bereich den SkripteKnoten.
4. Führen Sie eine der folgenden Aktionen aus:

- Um eine einzelne zu konvertierenFastExportJob-Skript, erweitern Sie dasFastExportKnoten, wählen Sie das zu konvertierende Skript aus und wählen Sie dannSkript konvertierenaus dem Kontextmenü (Rechtsklick).
- Um mehrere Skripts zu konvertieren, stellen Sie sicher, dass Sie alle zu konvertierenden Skripts auswählen. Dann wähleFastExport, öffnen Sie das Kontextmenü (Rechtsklick) und wählen SieSkript konvertieren.

AWS SCTkonvertiert alle Ihre ausgewählten TeradataFastExportJob-Skripts in einem Format, das mit Amazon Redshift RSQL kompatibel ist. Finden Sie Ihre konvertierten Skripte in derSkripteKnoten im Zieldatenbank-Panel.

5. Bearbeiten Sie Ihre konvertierten Amazon Redshift RSQL-Skripts oder speichern Sie sie. Weitere Informationen finden Sie unter [Bearbeiten und Speichern Ihrer konvertiertenFastExportJob-Skripte](#).

Verwaltung von TeradataFastExportJob-Skripte mitAWS SCT

Sie können mehrere Teradata hinzufügenFastExportJob-Skripte oder entferne eineFastExportJob-Skript von IhremAWS SCTprojekt.

Um ein neues hinzuzufügenFastExportJob-Skript zu IhremAWS SCTProjekt

1. Erweitern Sie dieSkripteKnoten im linken Bereich.
2. Wählen Sie dieFastExportKnoten, und öffnen Sie das Kontextmenü (Rechtsklick).
3. WähleSkripte laden.
4. Geben Sie die Informationen ein, die erforderlich sind, um ein neues hinzuzufügenFastExportJob-Skript und Konfiguration von Substitutionsvariablen. Weitere Informationen erhalten Sie unter [HinzufügenFastExportJob-Skripte zu IhremAWS SCTProjekt](#) und [Konfiguration von Substitutionsvariablen inFastExportJob-Skripte](#).

Um eine zu entfernenFastExportJob-Skript von IhremAWS SCTProjekt

1. Erweitern Sie dieFastExportKnoten unterSkripteim linken Bereich.
2. Wählen Sie das zu entfernende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. WähleSkript löschen.

Erstellen eines Bewertungsberichts für einen TeradataFastExportJob-Skript-Konvertierung mitAWS SCT

DerFastExportBericht zur Bewertung der Konvertierung von Jobskriptenbietet Informationen zur Konvertierung derFastExportBefehle und SQL-Anweisungen von IhremFastExportSkripts in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Der Bewertungsbericht enthält Maßnahmen fürFastExportBefehle und SQL-Anweisungen, dieAWS SCTkann nicht konvertieren.

So erstellen Sie einen Bericht zur Bewertung der Skriptkonvertierung für einen TeradataFastExportAufgabe

1. Erweitern Sie dieFastExportKnoten unterSkripteim linken Bereich.
2. Wählen Sie das zu konvertierende Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dannBericht erstellen.
3. Sehen Sie sich dieZusammenfassungTab. DerZusammenfassungAuf der Registerkarte werden die Informationen zur Zusammenfassung aus demFastExportBericht zur Bewertung des Job-Skripts. Es enthält Konvertierungsergebnisse für alleFastExportBefehle und SQL-Anweisungen aus Ihren Quellskripten.
4. Sie können eine lokale Kopie desFastExportBericht zur Bewertung der Konvertierung von Jobskripten entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV).
 - a. Um das zu rettenFastExportBericht zur Bewertung der Konvertierung von Jobskripten als PDF-Datei, wählen SieAls PDF speichernoben rechts.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Konvertierung von Skripten.

- b. Um das zu rettenFastExportBericht zur Bewertung der Jobskript-Konvertierung als CSV-Datei, wählen SieIn CSV speichernoben rechts.

Die CSV-Datei enthält Aktionselemente, empfohlene Aktionen und eine geschätzte Komplexität des manuellen Aufwands, der zur Konvertierung der Skripts erforderlich ist.

5. Wählen Sie dieAktionspunkteTab. Diese Registerkarte enthält eine Liste von Elementen, die eine manuelle Konvertierung zu Amazon Redshift RSQL erfordern. Wenn Sie ein Aktionselement aus der Liste auswählen,AWS SCThebt den Artikel aus Ihrer Quelle hervorFastExportJobskript, auf das sich das Aktionselement bezieht.

Bearbeiten und Speichern Ihrer konvertierten TeradataFastExportJob-Skripte mitAWS SCT

Sie können Ihre konvertierten Skripte im unteren Bereich IhresAWS SCTprojekt.AWS SCTspeichert das bearbeitete Skript als Teil Ihres Projekts.

Um Ihre konvertierten Skripte zu speichern

1. Erweitern Sie dieRSQL-SkriptsKnoten unterSkripteim Zieldatenbank-Panel.
2. Wählen Sie Ihr konvertiertes Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen SieSkript speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem das konvertierte Skript gespeichert werden soll, und wählen SieSpeichern.

AWS SCTspeichert das konvertierte Skript in einer Datei und öffnet diese Datei.

Teradata konvertierenFastLoadJobskripte für Amazon Redshift RSQL mitAWS SCT

Sie können das verwendenAWS Schema Conversion Tool(AWS SCT) um Teradata zu konvertierenFastLoadJobskripte für Amazon Redshift RSQL.

EINTeradataFastLoadDrehbuchist eine Reihe von Befehlen, die mehrere Sitzungen verwenden, um Daten in eine leere Tabelle in einer Teradata-Datenbank zu laden. TeradataFastLoadverarbeitet eine Reihe von TeradataFastLoadBefehle und SQL-Anweisungen. Die TeradataFastLoadBefehle ermöglichen die Sitzungssteuerung und die Datenverarbeitung der Datenübertragungen. Die SQL-Anweisungen erstellen, verwalten und löschen Tabellen.

AWS SCTkonvertiert TeradataFastLoadBefehle und SQL-Anweisungen in einem Format, das mit Amazon Redshift RSQL kompatibel ist. Nachdem Sie die Teradata-Datenbank zu Amazon Redshift migriert haben, können Sie diese konvertierten Skripts verwenden, um Daten in Ihre Amazon Redshift-Datenbank zu laden.

Themen

- [HinzufügenFastLoadJob-Skripte zu IhremAWS SCTProjekt](#)
- [Konfiguration von Substitutionsvariablen in TeradataFastLoadJob-Skripte mitAWS SCT](#)

- [Teradata konvertierenFastLoadJob-Skripte mitAWS SCT](#)
- [Verwaltung von TeradataFastLoadJob-Skripte mitAWS SCT](#)
- [Erstellen eines Bewertungsberichts für einen TeradataFastLoadJob-Script-Konvertierung mitAWS SCT](#)
- [Bearbeiten und Speichern Ihrer konvertierten TeradataFastLoadJob-Skripte mitAWS SCT](#)

HinzufügenFastLoadJob-Skripte zu IhremAWS SCTProjekt

Sie können mehrere Skripte zu einem einzigen hinzufügenAWS SCTprojekt.

Um eine hinzuzufügenFastLoadJob-Skript zu IhremAWS SCTProjekt

1. Erstellen Sie ein neues Projekt inAWS SCT, oder öffnen Sie ein vorhandenes Projekt. Weitere Informationen finden Sie unter [the section called “Erstellen eines Projekts”](#).
2. WähleQuelle hinzufügenaus dem Menü, und wählen Sie dannTeradataum Ihre Quelldatenbank zum Projekt hinzuzufügen. Weitere Informationen finden Sie unter [Teradata als Quelle verwenden](#).
3. WähleZiel hinzufügenaus dem Menü und fügen Sie eine Amazon Redshift-Zieldatenbank zu Ihrer hinzuAWS SCTprojekt.

Sie können eine virtuelle Amazon Redshift-Zieldatenbankplattform verwenden. Weitere Informationen finden Sie unter [Verwendung virtueller Ziele](#).

4. Erstellen Sie eine neue Zuordnungsregel, die Ihre Teradata-Quelldatenbank und Ihr Amazon Redshift-Ziel umfasst. Weitere Informationen finden Sie unter [Neue Zuordnungsregel hinzufügen](#).
5. Auf derAnsehenMenü, wählenHauptansicht.
6. Erweitern Sie im linken Bereich denSkripteKnoten.
7. WähleFastLoad, öffnen Sie das Kontextmenü (Rechtsklick) und wählen SieSkripte laden.
8. Geben Sie den Standort Ihrer Teradata-Quelle einFastLoadJob-Skripte und wähleOrdner wählen.

AWS SCTzeigt dieSkripte ladenFenster.

9. Führen Sie eine der folgenden Aktionen aus:
 - Wenn dein TeradataFastLoadJobskripte enthalten die Substitutionsvariablen nicht, wählen SieKeine Substitutionsvariablen, und wählen Sie dannOKAYum Skripte zu Ihrem hinzuzufügenAWS SCTprojekt.

- Wenn dein TeradataFastLoadJobskripte enthalten die Substitutionsvariablen, konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter [Konfiguration von Substitutionsvariablen inFastLoadJob-Skripte](#).

Konfiguration von Substitutionsvariablen in TeradataFastLoadJob-Skripte mitAWS SCT

Ihr TeradataFastLoadJobskripte können Substitutionsvariablen enthalten. Sie können beispielsweise ein einzelnes Skript mit Substitutionsvariablen verwenden, um Daten in verschiedene Datenbanken zu laden.

Bevor du einen läufstFastLoadJob-Skript mit Substitutionsvariablen, stellen Sie sicher, dass Sie die Werte für alle Variablen zuweisen. Dazu können Sie andere Tools oder Anwendungen wie ein Bash-Skript, UC4 (Automic) usw. verwenden.

AWS SCTkann Substitutionsvariablen erst auflösen und konvertieren, nachdem Sie ihre Werte zugewiesen haben. Bevor Sie mit der Konvertierung Ihres Quell-Teradata beginnenFastLoadJob-Skripts, stellen Sie sicher, dass Sie allen Substitutionsvariablen Werte zuweisen. Du kannst benutzenAWS SCTum Substitutionsvariablen in Ihren Teradata-Skripten zu konfigurieren.

Um Substitutionsvariablen in Ihrem zu konfigurierenFastLoadJob-Skript

1. Wenn Sie Ihre Teradata-Quelle hinzufügenFastLoadJob-Skripte zu IhremAWS SCTprojekt, wähleSubstitutionsvariablen werden verwendet. Weitere Hinweise zum Hinzufügen dieser Skripts finden Sie unter[HinzufügenFastLoadJob-Skripte zu IhremAWS SCTProjekt](#).
2. FürVariablenformat definieren, geben Sie einen regulären Ausdruck ein, der allen Substitutionsvariablen in Ihrem Skript entspricht.

Zum Beispiel, wenn die Namen Ihrer Substitutionsvariablen beginnen mit\${und ende mit}, benutze den $\backslash\$\{\backslashw+\}$ regulärer Ausdruck. Um Substitutionsvariablen abzugleichen, die entweder mit einem Dollarzeichen oder einem Prozentzeichen beginnen, verwenden Sie den $\backslash\$\backslashw+|\backslash%\backslashw+$ regulärer Ausdruck.

Reguläre Ausdrücke in AWS SCT entsprechen der Java-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie unter[java.util.regex Klassenmuster](#)in der Java-Dokumentation.

3. WähleOKAYum Skripte in Ihre zu ladenAWS SCTProjekt, und wählen Sie dannOKAYum das zu schließenSkripte ladenFenster.

4. Erweitern Sie im linken Bereich den SkripteKnoten. Wählen Sie FastLoad, und wählen Sie dann Ihren Ordner mit Skripten aus. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Variablen exportieren unter Substitutionsvariablen.

Sie können auch Substitutionsvariablen für ein Skript exportieren. Erweitern Sie Ihren Ordner mit Skripten, wählen Sie Ihr Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Variablen exportieren unter Substitutionsvariablen.

5. Geben Sie den Namen der Datei mit kommagetrennten Werten (CSV) ein, um die Substitutionsvariablen zu speichern, und wählen Sie dann Speichern.
6. Öffnen Sie diese CSV-Datei und geben Sie die Werte für die Substitutionsvariablen ein.

Je nach Betriebssystem AWS SCT verwendet verschiedene Formate für die CSV-Datei. Die Werte in der Datei können entweder in Anführungszeichen stehen oder nicht. Stellen Sie sicher, dass Sie für die Werte der Substitutionsvariablen dasselbe Format verwenden wie für die anderen Werte in der Datei. AWS SCT kann die CSV-Datei mit Werten in verschiedenen Formaten nicht importieren.

7. Speichern Sie die CSV-Datei.
8. Erweitern Sie im linken Bereich den SkripteKnoten. Wählen Sie FastLoad, und wählen Sie dann Ihr Skript aus. Öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dann Variablen importieren unter Substitutionsvariablen.
9. Wählen Sie Ihre CSV-Datei und wählen Sie dann Öffnen.
10. Wählen Sie Variablen um alle entdeckten Substitutionsvariablen und ihre Werte einzusehen.

Teradata konvertieren FastLoadJob-Skripte mit AWS SCT

Im Folgenden erfahren Sie, wie Sie Teradata konvertieren FastLoadJob zu Amazon Redshift RSQL mithilfe von AWS SCT.

Um einen Teradata zu konvertieren FastLoadJobskript für Amazon Redshift RSQL

1. Füge deine hinzu FastLoadJob-Skripte zu Ihrem AWS SCTprojekt. Weitere Informationen finden Sie unter [Hinzufügen FastLoadJob-Skripte zu Ihrem AWS SCTProjekt](#).
2. Konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter [Konfiguration von Substitutionsvariablen in FastLoadJob-Skripte](#).
3. Erweitern Sie im linken Bereich den SkripteKnoten.
4. Führen Sie eine der folgenden Aktionen aus:

- Um eine einzelne zu konvertierenFastLoadJob-Skript, erweitern Sie dasFastLoadKnoten, wählen Sie das zu konvertierende Skript aus und wählen Sie dannSkript konvertierenaus dem Kontextmenü (Rechtsklick).
- Um mehrere Skripts zu konvertieren, stellen Sie sicher, dass Sie alle zu konvertierenden Skripts auswählen. WähleFastLoad, öffnen Sie das Kontextmenü (Rechtsklick) und wählen SieSkript konvertieren. Führen Sie dann einen der folgenden Schritte aus:
 - Wenn Sie Ihre Quelldatendatei auf Amazon S3 speichern, wählen SieS3-ObjektpfadzumSpeicherort der Quelldatendatei.

Geben Sie Werte ein fürAmazon S3-Bucket-OrdnerundAmazon S3-Bucket für die Manifestdateifür Ihre Quelldatendatei.

- Wenn Sie Ihre Quelldatendatei nicht auf Amazon S3 speichern, wählen SieHostadressezumSpeicherort der Quelldatendatei.

Geben Sie Werte ein fürURL oder IP-Adresse des Hosts,Anmeldenname des Host-Benutzers, undAmazon S3-Bucket für die Manifestdateifür Ihre Quelldatendatei.

5. Wählen Sie OK.

AWS SCTkonvertiert alle Ihre ausgewählten TeradataFastLoadJob-Skripts in einem Format, das mit Amazon Redshift RSQL kompatibel ist. Finden Sie Ihre konvertierten Skripte in derSkripteKnoten im Zieldatenbank-Panel.

- ## 6. Bearbeiten Sie Ihre konvertierten Amazon Redshift RSQL-Skripts oder speichern Sie sie. Weitere Informationen finden Sie unter [Bearbeiten und Speichern Ihrer konvertiertenFastLoadJob-Skripte](#).

Verwaltung von TeradataFastLoadJob-Skripte mitAWS SCT

Sie können mehrere Teradata hinzufügenFastLoadJob-Skripte oder entferne eineFastLoadJob-Skript von IhremAWS SCTprojekt.

Um ein neues hinzuzufügenFastLoadJob-Skript zu IhremAWS SCTProjekt

1. Erweitern Sie dieSkripteKnoten im linken Bereich.
2. Wählen Sie dieFastLoadKnoten und öffnen Sie das Kontextmenü (Rechtsklick).
3. WähleSkripte laden.

4. Geben Sie die Informationen ein, die erforderlich sind, um ein neues hinzuzufügenFastLoadJob-Skript und Konfiguration von Substitutionsvariablen. Weitere Informationen erhalten Sie unter [HinzufügenFastLoadJob-Skripte zu IhremAWS SCTProjekt](#) und [Konfiguration von Substitutionsvariablen inFastLoadJob-Skripte](#).

Um eine zu entfernenFastLoadJob-Skript von IhremAWS SCTProjekt

1. Erweitern Sie dieFastLoadKnoten unterSkripteim linken Bereich.
2. Wählen Sie das zu entfernende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. WähleSkript löschen.

Erstellen eines Bewertungsberichts für einen TeradataFastLoadJob-Script-Konvertierung mitAWS SCT

DerFastLoadBericht zur Bewertung der Konvertierung von Jobskriptenbietet Informationen zur Konvertierung derFastLoadBefehle und SQL-Anweisungen. Die Konvertierung erfolgt von Ihren Quellskripten in ein Format, das mit Amazon Redshift RSQL kompatibel ist. Der Bewertungsbericht enthält Maßnahmen fürFastLoadBefehle und SQL-Anweisungen, dieAWS SCTkann nicht konvertieren.

So erstellen Sie einen Bericht zur Bewertung der Skriptkonvertierung für einen TeradataFastLoadAufgabe

1. Erweitern Sie dieFastLoadKnoten unterSkripteim linken Bereich.
2. Wählen Sie das zu konvertierende Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie dannBericht erstellen.
3. Sehen Sie sich dieZusammenfassungTab.

DerZusammenfassungAuf der Registerkarte werden die Informationen zur Zusammenfassung aus demFastLoadBericht zur Bewertung des Job-Skripts. Es enthält Konvertierungsergebnisse für alleFastLoadBefehle und SQL-Anweisungen aus Ihren Quellskripten.

4. (Optional) Speichern Sie eine lokale Kopie desFastLoadBericht zur Bewertung der Konvertierung von Job-Skripten entweder als PDF-Datei oder als Datei mit kommasetrennten Werten (CSV):
 - Um das zu rettenFastLoadBericht zur Bewertung der Konvertierung von Jobskripten als PDF-Datei, wählen SieAls PDF speichernoben rechts.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Skriptkonvertierung.

- Um das zu rettenFastLoadBericht zur Bewertung der Jobskript-Konvertierung als CSV-Datei, wählen SieIn CSV speichernoben rechts.

Die CSV-Datei enthält Aktionselemente, empfohlene Aktionen und eine geschätzte Komplexität des manuellen Aufwands, der zur Konvertierung der Skripts erforderlich ist.

5. Wählen Sie dieAktionspunkteTab. Diese Registerkarte enthält eine Liste von Elementen, die eine manuelle Konvertierung zu Amazon Redshift RSQL erfordern. Wenn Sie ein Aktionselement aus der Liste auswählen,AWS SCThebt den Artikel aus Ihrer Quelle hervorFastLoadJobskript, auf das sich das Aktionselement bezieht.

Bearbeiten und Speichern Ihrer konvertierten TeradataFastLoadJob-Skripte mitAWS SCT

Sie können Ihre konvertierten Skripte im unteren Bereich IhresAWS SCTprojekt.AWS SCTspeichert das bearbeitete Skript als Teil Ihres Projekts.

Um Ihre konvertierten Skripte zu speichern

1. Erweitern Sie dieRSQL-SkriptsKnoten unterSkripteim Zieldatenbank-Panel.
2. Wählen Sie Ihr konvertiertes Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen SieSkript speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem das konvertierte Skript gespeichert werden soll, und wählen SieSpeichern.

AWS SCTspeichert das konvertierte Skript in einer Datei und öffnet diese Datei.

Teradata konvertierenMultiLoadJobskripte für Amazon Redshift RSQL mitAWS SCT

Du kannst benutzenAWS SCTum Teradata zu konvertierenMultiLoadJobskripte für Amazon Redshift RSQL.

EINTeradataMultiLoadJob-Skript ist eine Reihe von Befehlen für die Batch-Wartung Ihrer Teradata-Datenbank. Ein TeradataMultiLoadDie Importaufgabe führt eine Reihe verschiedener Einfüge-, Aktualisierungs- und Löschvorgänge für bis zu fünf verschiedene Tabellen und Ansichten aus. TeradataMultiLoadAufgaben löschen kann eine große Anzahl von Zeilen aus einer einzelnen Tabelle entfernen.

AWS SCT konvertiert TeradataMultiLoadBefehle und SQL-Anweisungen in einem Format, das mit Amazon Redshift RSQL kompatibel ist. Nachdem Sie die Teradata-Datenbank zu Amazon Redshift migriert haben, verwenden Sie diese konvertierten Skripts, um Daten in Ihrer Amazon Redshift-Datenbank zu verwalten.

Themen

- [HinzufügenMultiLoadJob-Skripte zu IhremAWS SCTProjekt](#)
- [Konfiguration von Substitutionsvariablen in TeradataMultiLoadJob-Skripte mitAWS SCT](#)
- [Teradata konvertierenMultiLoadJob-Skripte mitAWS SCT](#)
- [Verwaltung von TeradataMultiLoadJob-Skripte mitAWS SCT](#)
- [Erstellen eines Bewertungsberichts für einen TeradataMultiLoadJob-Script-Konvertierung mitAWS SCT](#)
- [Bearbeiten und Speichern Ihrer konvertierten TeradataMultiLoadJob-Skripte mitAWS SCT](#)

HinzufügenMultiLoadJob-Skripte zu IhremAWS SCTProjekt

Sie können mehrere Skripte zu einem einzigen hinzufügenAWS SCTprojekt.

Um eine hinzuzufügenMultiLoadJob-Skript zu IhremAWS SCTProjekt

1. Erstellen Sie ein neues Projekt inAWS SCToder öffnen Sie ein bestehendes Projekt. Weitere Informationen finden Sie unter [the section called “Erstellen eines Projekts”](#).
2. WähleQuelle hinzufügenaus dem Menü, und wählen Sie dannTeradataum Ihre Quelldatenbank zum Projekt hinzuzufügen. Weitere Informationen finden Sie unter [Teradata als Quelle verwenden](#).
3. WähleZiel hinzufügenaus dem Menü und um eine Amazon Redshift-Zieldatenbank zu IhrerAWS SCTprojekt.

Sie können eine virtuelle Amazon Redshift-Zieldatenbankplattform verwenden. Weitere Informationen finden Sie unter [Verwendung virtueller Ziele](#).

4. Erstellen Sie eine neue Zuordnungsregel, die Ihre Teradata-Quelldatenbank und Ihr Amazon Redshift-Ziel umfasst. Weitere Informationen finden Sie unter [Neue Zuordnungsregel hinzufügen](#).
5. Auf der Ansehen-Menü, wählen Hauptansicht.
6. Erweitern Sie im linken Bereich den Skripte-Knoten.
7. Wähle MultiLoad, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Skripte laden.
8. Geben Sie den Standort Ihrer Teradata-Quelle ein MultiLoadJob-Skripte und wähle Ordner wählen.

AWS SCT zeigt die Skripte laden Fenster.

9. Führen Sie eine der folgenden Aktionen aus:
 - Wenn dein Teradata MultiLoadJob-Skripte enthalten die Substitutionsvariablen nicht, wählen Sie Keine Substitutionsvariablen, und wählen Sie dann OKAY um Skripte zu Ihrem hinzuzufügen AWS SCT projekt.
 - Wenn dein Teradata MultiLoadJob-Skripte enthalten die Substitutionsvariablen, konfigurieren Sie die Substitutionsvariablen. Weitere Informationen finden Sie unter [Konfiguration von Substitutionsvariablen in MultiLoadJob-Skripte](#).

Konfiguration von Substitutionsvariablen in Teradata MultiLoadJob-Skripte mit AWS SCT

Ihr Teradata MultiLoadJob-Skripte können Substitutionsvariablen enthalten. Sie können beispielsweise ein einzelnes Skript mit Substitutionsvariablen verwenden, um Daten in verschiedene Datenbanken zu laden.

Bevor du einen läufst MultiLoadJob-Skript mit Substitutionsvariablen, stellen Sie sicher, dass Sie die Werte für alle Variablen zuweisen. Dazu können Sie andere Tools oder Anwendungen wie ein Bash-Skript, UC4 (Automic) usw. verwenden.

AWS SCT kann Substitutionsvariablen erst auflösen und konvertieren, nachdem Sie ihre Werte zugewiesen haben. Bevor Sie mit der Konvertierung Ihres Quell-Teradata beginnen MultiLoadJob-Skripts, stellen Sie sicher, dass Sie allen Substitutionsvariablen Werte zugewiesen haben. Du kannst benutzen AWS SCT um Substitutionsvariablen in Ihren Teradata-Skripten zu konfigurieren.

Um Substitutionsvariablen in Ihrem zu konfigurierenMultiLoadJob-Skript

1. Wenn Sie Ihre Teradata-Quelle hinzufügenMultiLoadJob-Skripte zu IhremAWS SCTprojekt, wähleSubstitutionsvariablen werden verwendet. Weitere Hinweise zum Hinzufügen dieser Skripts finden Sie unter[HinzufügenMultiLoadJob-Skripte zu IhremAWS SCTProjekt](#).
2. FürVariablenformat definieren, geben Sie einen regulären Ausdruck ein, der allen Substitutionsvariablen in Ihrem Skript entspricht.

Zum Beispiel, wenn die Namen Ihrer Substitutionsvariablen beginnen mit\${und ende mit}, benutze den\\$\{\w+\}regulärer Ausdruck. Um Substitutionsvariablen abzugleichen, die entweder mit einem Dollarzeichen oder einem Prozentzeichen beginnen, verwenden Sie den\\$\w+|\%\w+regulärer Ausdruck.

Reguläre Ausdrücke in AWS SCT entsprechen der Java-Syntax für reguläre Ausdrücke. Weitere Informationen finden Sie unter[java.util.regex Klassenmuster](#)in der Java-Dokumentation.

3. WähleOKAYum Skripte in Ihre zu ladenAWS SCTProjekt, und wählen Sie dannOKAYum das zu schließenSkripte ladenFenster.
4. WähleVariablenum alle entdeckten Substitutionsvariablen und ihre Werte einzusehen.
5. FürWert, geben Sie den Wert für die Substitutionsvariable ein.

Teradata konvertierenMultiLoadJob-Skripte mitAWS SCT

Im Folgenden erfahren Sie, wie Sie Teradata konvertierenMultiLoadJob zu Amazon Redshift RSQLe mithilfe vonAWS SCT.

Um einen Teradata zu konvertierenMultiLoadJobskript für Amazon Redshift RSQLe

1. Füge deine hinzuMultiLoadJob-Skripte zu IhremAWS SCTprojekt. Weitere Informationen finden Sie unter [HinzufügenMultiLoadJob-Skripte zu IhremAWS SCTProjekt](#).
2. Konfigurieren Sie die Substitutionsvariablen und geben Sie ihre Werte ein. Weitere Informationen finden Sie unter [Konfiguration von Substitutionsvariablen inMultiLoadJob-Skripte](#).
3. Erweitern Sie im linken Bereich denSkripteKnoten.
4. Führen Sie eine der folgenden Aktionen aus:
 - Um eine einzelne zu konvertierenMultiLoadJob-Skript, erweitern Sie dasMultiLoadKnoten, wählen Sie das zu konvertierende Skript aus und wählen Sie dannSkript konvertierenaus dem Kontextmenü (Rechtsklick).

- Um mehrere Skripts zu konvertieren, stellen Sie sicher, dass Sie alle zu konvertierenden Skripts auswählen. Wählen Sie MultiLoad, öffnen Sie das Kontextmenü (Rechtsklick) und wählen Sie Skript konvertieren.
5. Führen Sie eine der folgenden Aktionen aus:
- Wenn Sie Ihre Quelldatendatei auf Amazon S3 speichern, wählen Sie S3-Objektpfad zum Speicherort der Quelldatendatei.

Eingeben Amazon S3-Bucket-Ordner und Amazon S3-Bucket für die Manifestdatei für Ihre Quelldatendatei.
 - Wenn Sie Ihre Quelldatendatei nicht auf Amazon S3 speichern, wählen Sie Hostadresse zum Speicherort der Quelldatendatei.

Eingeben URL oder IP-Adresse des Hosts, Anmeldenname des Host-Benutzers, und Amazon S3-Bucket für die Manifestdatei für Ihre Quelldatendatei.
6. Wählen Sie OK.
- AWS SCT konvertiert alle Ihre ausgewählten Teradata MultiLoadJob-Skripts in einem Format, das mit Amazon Redshift RSQL kompatibel ist. Finden Sie Ihre konvertierten Skripte in der SkripteKnoten im Zieldatenbank-Panel.
7. Bearbeiten Sie Ihre konvertierten Amazon Redshift RSQL-Skripts oder speichern Sie sie. Weitere Informationen finden Sie unter [Bearbeiten und Speichern Ihrer konvertierten MultiLoadJob-Skripte](#).

Verwaltung von Teradata MultiLoadJob-Skripte mit AWS SCT

Sie können mehrere Teradata hinzufügen MultiLoadJob-Skripte oder entferne eine MultiLoadJob-Skript von Ihrem AWS SCT projekt.

Um ein neues hinzuzufügen MultiLoadJob-Skript zu Ihrem AWS SCT Projekt

1. Erweitern Sie die SkripteKnoten im linken Bereich.
2. Wählen Sie die MultiLoadKnoten und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie Skripte laden.
4. Geben Sie die Informationen ein, die erforderlich sind, um ein neues hinzuzufügen MultiLoadJob-Skript und Konfiguration von Substitutionsvariablen. Weitere Informationen erhalten Sie

unter [HinzufügenMultiLoadJob-Skripte zu IhremAWS SCTProjekt](#) und [Konfiguration von Substitutionsvariablen inMultiLoadJob-Skripte](#).

Um eine zu entfernenMultiLoadJob-Skript von IhremAWS SCTProjekt

1. Erweitern Sie dieMultiLoadKnoten unterSkripteim linken Bereich.
2. Wählen Sie das zu entfernende Skript aus und öffnen Sie das Kontextmenü (Rechtsklick).
3. WähleSkript löschen.

Erstellen eines Bewertungsberichts für einen TeradataMultiLoadJob-Script-Konvertierung mitAWS SCT

DerMultiLoadBericht zur Bewertung der Konvertierung von Jobskriptenbietet Informationen zur Konvertierung derMultiLoadBefehle und SQL-Anweisungen. Die Konvertierung erfolgt von Ihren Quellskripten zu Amazon Redshift RSQL-Befehlen und SQL-Anweisungen für Amazon Redshift. Der Bewertungsbericht enthält Maßnahmen fürMultiLoadBefehle und SQL-Anweisungen, dieAWS SCTkann nicht konvertieren.

So erstellen Sie einen Bericht zur Bewertung der Skriptkonvertierung für einen TeradataMultiLoadAufgabe

1. Erweitern Sie dieMultiLoadKnoten unterSkripteim linken Bereich.
2. Wählen Sie die Skripts aus, für die der Bewertungsbericht erstellt werden soll, öffnen Sie das Kontextmenü (Rechtsklick), und wählen Sie dannBericht erstellen.
3. Sehen Sie sich dieZusammenfassungTab. DerZusammenfassungAuf der Registerkarte werden die Informationen zur Zusammenfassung aus demMultiLoadBericht zur Bewertung des Job-Skripts. Es enthält Konvertierungsergebnisse für alleMultiLoadBefehle und SQL-Anweisungen aus Ihren Quellskripten.
4. (Optional) Speichern Sie eine lokale Kopie desMultiLoadBericht zur Bewertung der Konvertierung von Job-Skripten entweder als PDF-Datei oder als CSV-Datei (durch Kommas getrennte Werte):
 - Um das zu rettenMultiLoadBericht zur Bewertung der Konvertierung von Jobskripten als PDF-Datei, wählen SieAls PDF speichernoben rechts.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Konvertierung von Skripten.

- Um das zu rettenMultiLoadBericht zur Bewertung der Jobskript-Konvertierung als CSV-Datei, wählen SieIn CSV speichernoben rechts.

AWS SCTerstellt zwei CSV-Dateien. Diese Dateien enthalten die Zusammenfassung, Aktionspunkte, Handlungsempfehlungen und eine geschätzte Komplexität des manuellen Aufwands, der zur Konvertierung der Skripts erforderlich ist.

5. Wählen Sie dieAktionspunkteTab. Diese Registerkarte enthält eine Liste von Elementen, die eine manuelle Konvertierung zu Amazon Redshift RSQL erfordern. Wenn Sie ein Aktionselement aus der Liste auswählen,AWS SCThebt den Artikel aus Ihrer Quelle hervorMultiLoadJobskript, auf das sich das Aktionselement bezieht.

Bearbeiten und Speichern Ihrer konvertierten TeradataMultiLoadJob-Skripte mitAWS SCT

Sie können Ihre konvertierten Skripte im unteren Bereich IhresAWS SCTprojekt.AWS SCTspeichert das bearbeitete Skript als Teil Ihres Projekts.

Um Ihre konvertierten Skripte zu speichern

1. Erweitern Sie dieRSQL-SkriptsKnoten unterSkripteim Zieldatenbank-Panel.
2. Wählen Sie Ihr konvertiertes Skript aus, öffnen Sie das Kontextmenü (Rechtsklick) und wählen SieSkript speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem das konvertierte Skript gespeichert werden soll, und wählen SieSpeichern.

AWS SCTspeichert das konvertierte Skript in einer Datei und öffnet diese Datei.

Migration von Big-Data-Frameworks mit AWS Schema Conversion Tool

Sie können das AWS Schema Conversion Tool (AWS SCT) verwenden, um Big-Data-Frameworks auf die AWS Cloud zu migrieren.

AWS SCT unterstützt derzeit die Migration von Hadoop-Clustern zu Amazon EMR und Amazon S3. Dieser Migrationsprozess umfasst Hive- und HDFS-Dienste.

Sie können es auch verwenden, AWS SCT um die Konvertierung Ihrer Apache Oozie Orchestrierungsworkflows zu automatisieren. AWS Step Functions

Themen

- [Migration von Apache Hadoop zu Amazon EMR mit dem AWS Schema Conversion Tool](#)
- [Konvertierung von Apache Oozie zu AWS Step Functions mit dem AWS Schema Conversion Tool](#)

Migration von Apache Hadoop zu Amazon EMR mit dem AWS Schema Conversion Tool

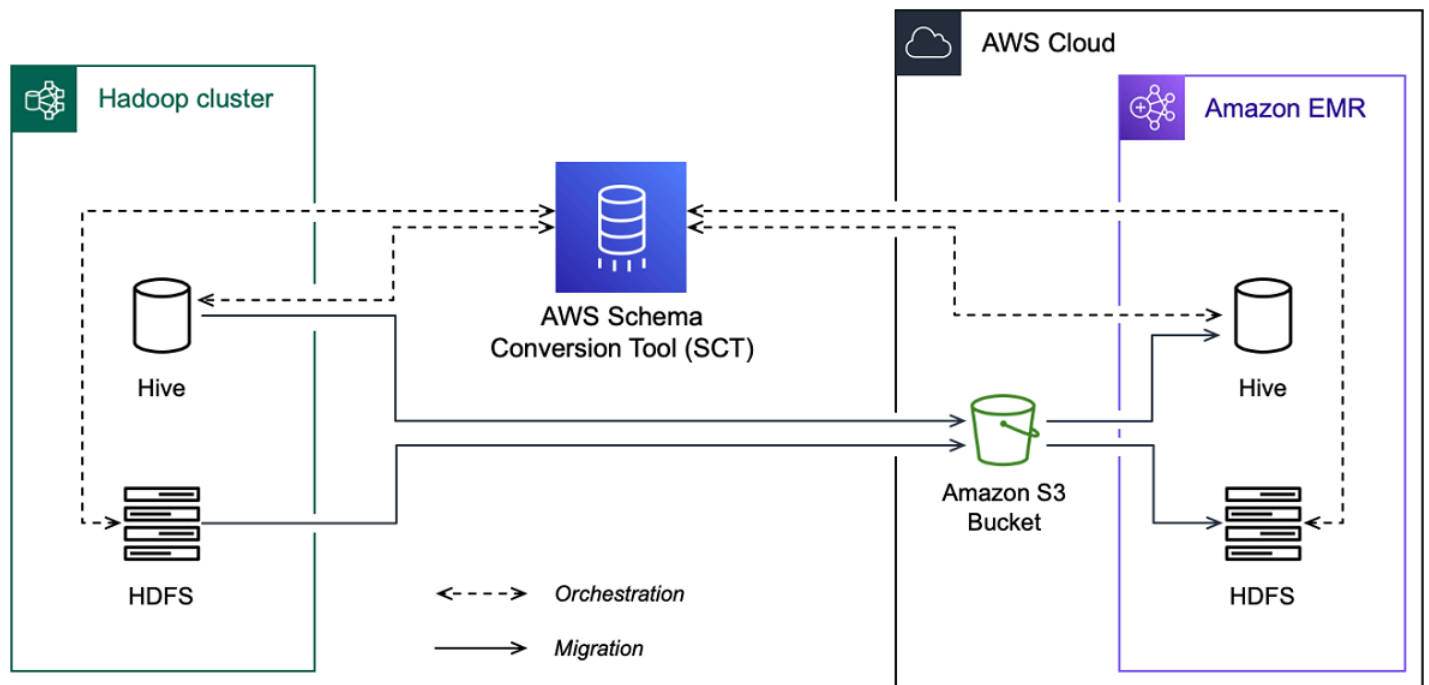
Um Apache Hadoop-Cluster zu migrieren, stellen Sie sicher, dass Sie AWS SCT Version 1.0.670 oder höher verwenden. Machen Sie sich auch mit der Befehlszeilenschnittstelle (CLI) von AWS SCT vertraut. Weitere Informationen finden Sie unter [AWS SCT CLI-Referenz](#).

Themen

- [Überblick über die Migration](#)
- [Schritt 1: Stellen Sie eine Verbindung zu Ihren Hadoop-Clustern her](#)
- [Schritt 2: Richten Sie die Zuordnungsregeln ein](#)
- [Schritt 3: Erstellen Sie einen Bewertungsbericht](#)
- [Schritt 4: Migrieren Sie Ihren Apache Hadoop-Cluster zu Amazon EMR mit AWS SCT](#)
- [Ihr CLI-Skript ausführen](#)
- [Verwaltung Ihres Big-Data-Migrationsprojekts](#)

Überblick über die Migration

Die folgende Abbildung zeigt das Architekturdiagramm der Migration von Apache Hadoop zu Amazon EMR.



AWS SCT migriert Daten und Metadaten von Ihrem Hadoop-Quellcluster in einen Amazon S3-Bucket. Verwenden Sie anschließend AWS SCT Ihre Hive-Quellmetadaten, um Datenbankobjekte im Amazon EMR Hive-Zieldienst zu erstellen. Optional können Sie Hive so konfigurieren, dass es AWS Glue Data Catalog als Metastore verwendet wird. In diesem Fall AWS SCT migriert Ihre Hive-Quellmetadaten in die AWS Glue Data Catalog

Anschließend können AWS SCT Sie die Daten aus einem Amazon S3-Bucket zu Ihrem Amazon EMR HDFS-Zieldienst migrieren. Alternativ können Sie die Daten in Ihrem Amazon S3-Bucket belassen und sie als Datenrepository für Ihre Hadoop-Workloads verwenden.

Um die Hadoop-Migration zu starten, erstellen Sie Ihr AWS SCT CLI-Skript und führen es aus. Dieses Skript enthält den kompletten Befehlssatz zum Ausführen der Migration. Sie können eine Vorlage des Hadoop-Migrationsskripts herunterladen und bearbeiten. Weitere Informationen finden Sie unter [CLI-Szenarien abrufen](#).

Stellen Sie sicher, dass Ihr Skript die folgenden Schritte enthält, damit Sie Ihre Migration von Apache Hadoop zu Amazon S3 und Amazon EMR ausführen können.

Schritt 1: Stellen Sie eine Verbindung zu Ihren Hadoop-Clustern her

Erstellen Sie ein neues AWS SCT Projekt, um die Migration Ihres Apache Hadoop-Clusters zu starten. Stellen Sie als Nächstes eine Verbindung zu Ihren Quell- und Zielclustern her. Stellen Sie sicher, dass Sie Ihre AWS Zielressourcen erstellen und bereitstellen, bevor Sie mit der Migration beginnen.

In diesem Schritt verwenden Sie die folgenden AWS SCT CLI-Befehle.

- `CreateProject`— um ein neues AWS SCT Projekt zu erstellen.
- `AddSourceCluster`— um eine Verbindung zum Hadoop-Quellcluster in Ihrem AWS SCT Projekt herzustellen.
- `AddSourceClusterHive`— um eine Verbindung zum Hive-Quelldienst in Ihrem Projekt herzustellen.
- `AddSourceClusterHDFS`— um eine Verbindung zum HDFS-Quelldienst in Ihrem Projekt herzustellen.
- `AddTargetCluster`— um eine Verbindung zum Amazon EMR-Zielcluster in Ihrem Projekt herzustellen.
- `AddTargetClusterS3`— um den Amazon S3-Bucket zu Ihrem Projekt hinzuzufügen.
- `AddTargetClusterHive`— um eine Verbindung zum Ziel-Hive-Dienst in Ihrem Projekt herzustellen
- `AddTargetClusterHDFS`— um eine Verbindung zum HDFS-Zieldienst in Ihrem Projekt herzustellen

Beispiele für die Verwendung dieser AWS SCT CLI-Befehle finden Sie unter [Apache Hadoop als Quelle verwenden](#).

Wenn Sie den Befehl ausführen, der eine Verbindung zu einem Quell- oder Zielcluster herstellt, wird AWS SCT versucht, die Verbindung zu diesem Cluster herzustellen. Wenn der Verbindungsversuch fehlschlägt, wird die Ausführung der Befehle aus Ihrem CLI-Skript AWS SCT beendet und eine Fehlermeldung angezeigt.

Schritt 2: Richten Sie die Zuordnungsregeln ein

Nachdem Sie eine Verbindung zu Ihren Quell- und Zielclustern hergestellt haben, richten Sie die Zuordnungsregeln ein. Eine Zuordnungsregel definiert das Migrationsziel für einen Quellcluster.

Stellen Sie sicher, dass Sie Zuordnungsregeln für alle Quellcluster einrichten, die Sie Ihrem AWS SCT Projekt hinzugefügt haben. Weitere Hinweise zu Zuordnungsregeln finden Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#).

In diesem Schritt verwenden Sie den `AddServerMapping` Befehl. Dieser Befehl verwendet zwei Parameter, die die Quell- und Zielcluster definieren. Sie können den `AddServerMapping` Befehl mit dem expliziten Pfad zu Ihren Datenbankobjekten oder mit einem Objektnamen verwenden. Für die erste Option geben Sie den Typ des Objekts und seinen Namen an. Bei der zweiten Option geben Sie nur die Objektnamen an.

- `sourceTreePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten.
`targetTreePath`— der explizite Pfad zu Ihren Zieldatenbankobjekten.
- `sourceNamePath`— der Pfad, der nur die Namen Ihrer Quellobjekte enthält.
`targetNamePath`— der Pfad, der nur die Namen Ihrer Zielobjekte enthält.

Im folgenden Codebeispiel wird eine Zuordnungsregel erstellt, die explizite Pfade für die `testdb` Quell-Hive-Datenbank und den EMR-Zielcluster verwendet.

```
AddServerMapping
  -sourceTreePath: 'Clusters.HADOOP_SOURCE.HIVE_SOURCE.Databases.testdb'
  -targetTreePath: 'Clusters.HADOOP_TARGET.HIVE_TARGET'
/
```

Sie können dieses Beispiel und die folgenden Beispiele in Windows verwenden. Um die CLI-Befehle unter Linux auszuführen, stellen Sie sicher, dass Sie die Dateipfade für Ihr Betriebssystem entsprechend aktualisiert haben.

Im folgenden Codebeispiel wird eine Zuordnungsregel erstellt, die die Pfade verwendet, die nur die Objektnamen enthalten.

```
AddServerMapping
  -sourceNamePath: 'HADOOP_SOURCE.HIVE_SOURCE.testdb'
  -targetNamePath: 'HADOOP_TARGET.HIVE_TARGET'
/
```

Sie können Amazon EMR oder Amazon S3 als Ziel für Ihr Quellobjekt wählen. Für jedes Quellobjekt können Sie in einem einzelnen AWS SCT Projekt nur ein Ziel auswählen. Um das Migrationsziel

für ein Quellobjekt zu ändern, löschen Sie die vorhandene Zuordnungsregel und erstellen Sie dann eine neue Zuordnungsregel. Verwenden Sie den `DeleteServerMapping` Befehl, um eine Zuordnungsregel zu löschen. Dieser Befehl verwendet einen der beiden folgenden Parameter.

- `sourceTreePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten.
- `sourceNamePath`— der Pfad, der nur die Namen Ihrer Quellobjekte enthält.

Weitere Informationen zu den `DeleteServerMapping` Befehlen `AddServerMapping` und finden Sie in der [AWS Schema Conversion Tool CLI-Referenz](#).

Schritt 3: Erstellen Sie einen Bewertungsbericht

Bevor Sie mit der Migration beginnen, empfehlen wir, einen Bewertungsbericht zu erstellen. Dieser Bericht fasst alle Migrationsaufgaben zusammen und beschreibt detailliert die Aktionspunkte, die während der Migration anfallen werden. Um sicherzustellen, dass Ihre Migration nicht fehlschlägt, sehen Sie sich diesen Bericht an und gehen Sie vor der Migration auf die Aktionspunkte ein. Weitere Informationen finden Sie unter [Berichte über die Migrationsbewertung](#).

In diesem Schritt verwenden Sie den `CreateMigrationReport` Befehl. Dieser Befehl verwendet zwei Parameter. Der `treePath` Parameter ist obligatorisch und der `forceMigrate` Parameter ist optional.

- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `forceMigrate`— wenn auf `gesetzttrue`, AWS SCT wird die Migration auch dann fortgesetzt, wenn Ihr Projekt einen HDFS-Ordner und eine Hive-Tabelle enthält, die auf dasselbe Objekt verweisen. Der Standardwert ist `false`.

Anschließend können Sie eine Kopie des Bewertungsberichts als PDF oder als Datei mit kommagetrennten Werten (CSV) speichern. Verwenden Sie dazu den `SaveReportCSV` Befehl `SaveReportPDF` oder.

Der `SaveReportPDF` Befehl speichert eine Kopie Ihres Bewertungsberichts als PDF-Datei. Dieser Befehl verwendet vier Parameter. Der `file` Parameter ist obligatorisch, andere Parameter sind optional.

- `file`— der Pfad zur PDF-Datei und ihr Name.

- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Umfang Ihrer zu migrierenden Quellobjekte zu definieren.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `namePath`— der Pfad, der nur die Namen Ihrer Zielobjekte enthält, für die Sie eine Kopie des Bewertungsberichts speichern.

Der `SaveReportCSV` Befehl speichert Ihren Bewertungsbericht in drei CSV-Dateien. Dieser Befehl verwendet vier Parameter. Der `directory` Parameter ist obligatorisch, andere Parameter sind optional.

- `directory`— der Pfad zu dem Ordner, in dem die CSV-Dateien AWS SCT gespeichert sind.
- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Umfang Ihrer zu migrierenden Quellobjekte zu definieren.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `namePath`— der Pfad, der nur die Namen Ihrer Zielobjekte enthält, für die Sie eine Kopie des Bewertungsberichts speichern.

Im folgenden Codebeispiel wird eine Kopie des Bewertungsberichts in der `c:\sct\ar.pdf` Datei gespeichert.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

Im folgenden Codebeispiel wird eine Kopie des Bewertungsberichts als CSV-Datei im `c:\sct` Ordner gespeichert.

```
SaveReportCSV
-file:'c:\sct'
/
```

Weitere Informationen zu den `SaveReportCSV` Befehlen `SaveReportPDF` und finden Sie in der [AWS Schema Conversion Tool CLI-Referenz](#).

Schritt 4: Migrieren Sie Ihren Apache Hadoop-Cluster zu Amazon EMR mit AWS SCT

Nachdem Sie Ihr AWS SCT Projekt konfiguriert haben, beginnen Sie mit der Migration Ihres lokalen Apache Hadoop-Clusters zum AWS Cloud.

In diesem Schritt verwenden Sie die `ResumeMigration` Befehle `MigrateMigrationStatus`, und.

Der `Migrate` Befehl migriert Ihre Quellobjekte zum Zielcluster. Dieser Befehl verwendet vier Parameter. Stellen Sie sicher, dass Sie den `treePath` Parameter `filter` oder angeben. Andere Parameter sind optional.

- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Umfang Ihrer zu migrierenden Quellobjekte zu definieren.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `forceLoad`— Wenn auf `gesetztrue`, AWS SCT werden Datenbank-Metadatenbäume während der Migration automatisch geladen. Der Standardwert ist `false`.
- `forceMigrate`— wenn auf `gesetztrue`, AWS SCT wird die Migration auch dann fortgesetzt, wenn Ihr Projekt einen HDFS-Ordner und eine Hive-Tabelle enthält, die auf dasselbe Objekt verweisen. Der Standardwert ist `false`.

Der `MigrationStatus` Befehl gibt die Informationen über den Migrationsfortschritt zurück. Um diesen Befehl auszuführen, geben Sie den Namen Ihres Migrationsprojekts für den `name` Parameter ein. Sie haben diesen Namen im `CreateProject` Befehl angegeben.

Der `ResumeMigration` Befehl setzt die unterbrochene Migration fort, die Sie mit dem `Migrate` Befehl gestartet haben. Der `ResumeMigration` Befehl verwendet keine Parameter. Um die Migration fortzusetzen, müssen Sie eine Verbindung zu Ihren Quell- und Zielclustern herstellen. Weitere Informationen finden Sie unter [Verwaltung Ihres Migrationsprojekts](#).

Das folgende Codebeispiel migriert Daten von Ihrem HDFS-Quelldienst zu Amazon EMR.

```
Migrate
  -treePath: 'Clusters.HADOOP_SOURCE.HDFS_SOURCE'
  -forceMigrate: 'true'
/
```

Ihr CLI-Skript ausführen

Nachdem Sie die Bearbeitung Ihres AWS SCT CLI-Skripts abgeschlossen haben, speichern Sie es als Datei mit der `.scts` Erweiterung. Jetzt können Sie Ihr Skript aus dem `app` Ordner Ihres AWS SCT Installationspfads ausführen. Führen Sie dazu den folgenden Befehl aus.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\hadoop.scts"
```

Ersetzen Sie im vorherigen Beispiel *script_path durch den Pfad* zu Ihrer Datei mit dem CLI-Skript. Weitere Hinweise zum Ausführen von CLI-Skripten in finden Sie AWS SCT unter [Skriptmodus](#).

Verwaltung Ihres Big-Data-Migrationsprojekts

Nachdem Sie die Migration abgeschlossen haben, können Sie Ihr AWS SCT Projekt für die zukünftige Verwendung speichern und bearbeiten.

Verwenden Sie den `SaveProject` Befehl, um Ihr AWS SCT Projekt zu speichern. Dieser Befehl verwendet keine Parameter.

Das folgende Codebeispiel speichert Ihr AWS SCT Projekt.

```
SaveProject  
/
```

Verwenden Sie den `OpenProject` Befehl, um Ihr AWS SCT Projekt zu öffnen. Dieser Befehl verwendet einen obligatorischen Parameter. Geben Sie für den `file` Parameter den Pfad zu Ihrer AWS SCT Projektdatei und deren Namen ein. Sie haben den Projektnamen im `CreateProject` Befehl angegeben. Stellen Sie sicher, dass Sie die `.scts` Erweiterung zum Namen Ihrer Projektdatei hinzufügen, um den `OpenProject` Befehl auszuführen.

Das folgende Codebeispiel öffnet das `hadoop_emr` Projekt aus dem `c:\sct` Ordner.

```
OpenProject  
-file: 'c:\sct\hadoop_emr.scts'  
/
```

Nachdem Sie Ihr AWS SCT Projekt geöffnet haben, müssen Sie die Quell- und Zielcluster nicht hinzufügen, da Sie sie bereits zu Ihrem Projekt hinzugefügt haben. Um mit Ihren Quell- und Zielclustern arbeiten zu können, müssen Sie eine Verbindung zu ihnen herstellen. Dazu verwenden

Sie die `ConnectTargetCluster` Befehle `ConnectSourceCluster` und. Diese Befehle verwenden dieselben Parameter wie die `AddTargetCluster` Befehle `AddSourceCluster` und. Sie können Ihr CLI-Skript bearbeiten und den Namen dieser Befehle ersetzen, sodass die Liste der Parameter unverändert bleibt.

Das folgende Codebeispiel stellt eine Verbindung zum Hadoop-Quellcluster her.

```
ConnectSourceCluster
  -name: 'HADOOP_SOURCE'
  -vendor: 'HADOOP'
  -host: 'hadoop_address'
  -port: '22'
  -user: 'hadoop_user'
  -password: 'hadoop_password'
  -useSSL: 'true'
  -privateKeyPath: 'c:\path\name.pem'
  -passPhrase: 'hadoop_passphrase'
/
```

Das folgende Codebeispiel stellt eine Verbindung zum Amazon EMR-Zielcluster her.

```
ConnectTargetCluster
  -name: 'HADOOP_TARGET'
  -vendor: 'AMAZON_EMR'
  -host: 'ec2-44-44-55-66.eu-west-1.EXAMPLE.amazonaws.com'
  -port: '22'
  -user: 'emr_user'
  -password: 'emr_password'
  -useSSL: 'true'
  -privateKeyPath: 'c:\path\name.pem'
  -passPhrase: '1234567890abcdef0!'
  -s3Name: 'S3_TARGET'
  -accessKey: 'AKIAIOSFODNN7EXAMPLE'
  -secretKey: 'wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY'
  -region: 'eu-west-1'
  -s3Path: 'doc-example-bucket/example-folder'
/
```

Ersetzen Sie im vorherigen Beispiel *hadoop_address* durch die IP-Adresse Ihres Hadoop-Clusters. Konfigurieren Sie bei Bedarf den Wert der Portvariablen. Als Nächstes ersetzen Sie *hadoop_user* und *hadoop_password* durch den Namen Ihres Hadoop-Benutzers und

das Passwort für diesen Benutzer. Geben Sie für *path\ name* den Namen und den Pfad zur PEM-Datei für Ihren Hadoop-Quellcluster ein. Weitere Informationen zum Hinzufügen Ihrer Quell- und Zielcluster finden Sie unter [Verwendung von Apache Hadoop als Quelle für AWS SCT](#).

Nachdem Sie eine Verbindung zu Ihren Quell- und Ziel-Hadoop-Clustern hergestellt haben, müssen Sie eine Verbindung zu Ihren Hive- und HDFS-Diensten sowie zu Ihrem Amazon S3-Bucket herstellen. Dazu verwenden Sie die `ConnectTargetClusterS3` Befehle `ConnectSourceClusterHive`, `ConnectSourceClusterHdfs`, `ConnectTargetClusterHive`, `ConnectTargetClusterHdfs` und `ConnectTargetClusterS3`. Diese Befehle verwenden dieselben Parameter wie die Befehle, mit denen Sie Ihrem Projekt Hive- und HDFS-Dienste sowie den Amazon S3-Bucket hinzugefügt haben. Bearbeiten Sie das CLI-Skript, um das `Add` Präfix durch `Connect` in den Befehlsnamen zu ersetzen.

Konvertierung von Apache Oozie zu AWS Step Functions mit dem AWS Schema Conversion Tool

Um Apache Oozie-Workflows zu konvertieren, stellen Sie sicher, dass Sie AWS SCT Version 1.0.671 oder höher verwenden. Machen Sie sich auch mit der Befehlszeilenschnittstelle (CLI) von AWS SCT vertraut. Weitere Informationen finden Sie unter [AWS SCT CLI-Referenz](#).

Themen

- [Übersicht über die Konvertierung](#)
- [Schritt 1: Stellen Sie eine Verbindung zu Ihren Quell- und Zieldiensten her](#)
- [Schritt 2: Richten Sie die Zuordnungsregeln ein](#)
- [Schritt 3: Parameter konfigurieren](#)
- [Schritt 4: Erstellen Sie einen Bewertungsbericht](#)
- [Schritt 5: Konvertieren Sie Ihre Apache Oozie-Workflows auf mit AWS Step Functions AWS SCT](#)
- [Ihr CLI-Skript ausführen](#)
- [Apache Oozie-Knoten, die konvertiert werden können in AWS Step Functions](#)

Übersicht über die Konvertierung

Ihr Apache Oozie-Quellcode umfasst Aktionsknoten, Kontrollflussknoten und Jobeigenschaften. Aktionsknoten definieren die Jobs, die Sie in Ihrem Apache Oozie-Workflow ausführen. Wenn Sie Apache Oozie verwenden, um Ihren Apache Hadoop-Cluster zu orchestrieren, enthält ein Aktionsknoten einen Hadoop-Job. Kontrollflussknoten bieten einen Mechanismus zur Steuerung

des Workflow-Pfads. Zu den Kontrollflussknoten gehören Knoten wie `startend`, `decision`, `fork`, und `join`.

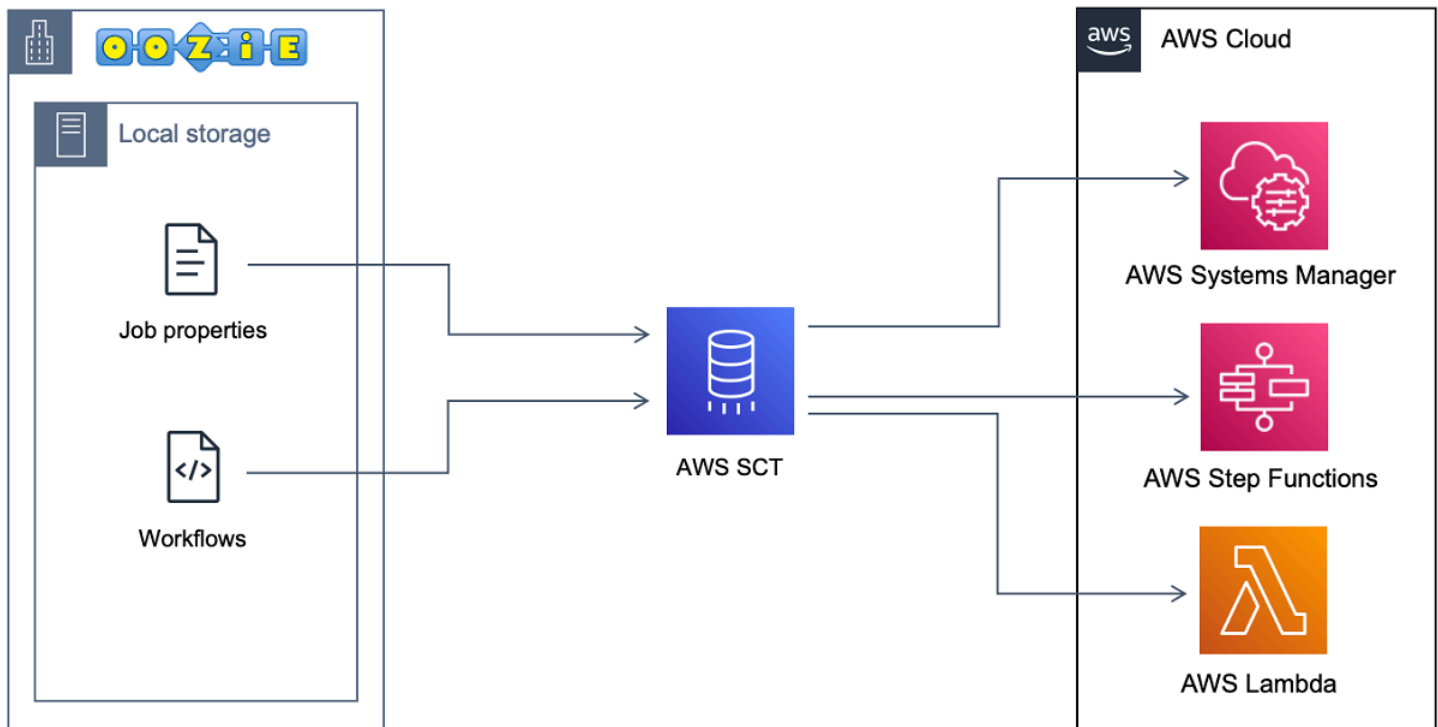
AWS SCT konvertiert Ihre Quell-Action-Knoten und Control-Flow-Knoten in AWS Step Functions. In AWS Step Functions definieren Sie Ihre Workflows in der Amazon States Language (ASL). AWS SCT verwendet ASL, um Ihre Zustandsmaschine zu definieren, eine Sammlung von Zuständen, die arbeiten können, bestimmen, zu welchen Zuständen als nächstes übergegangen werden soll, mit einem Fehler aufhören und so weiter. Als Nächstes werden die JSON-Dateien mit den Definitionen von State Machines AWS SCT hochgeladen. Anschließend AWS SCT können Sie Ihre AWS Identity and Access Management (IAM-) Rolle verwenden, um Ihre State-Maschinen in AWS Step Functions zu konfigurieren. Weitere Informationen finden Sie unter [Was ist AWS Step Functions?](#) im AWS Step Functions-Entwicklerhandbuch.

AWS SCT erzeugt außerdem ein Erweiterungspaket mit AWS Lambda Funktionen, die die Quellfunktionen emulieren, die AWS Step Functions nicht unterstützt werden. Weitere Informationen finden Sie unter [Verwenden von AWS SCT Erweiterungspaketen](#).

AWS SCT migriert Ihre Quelljob-Eigenschaften zu AWS Systems Manager AWS SCT verwendet Parameter Store, eine Funktion von, um Parameternamen und -werte zu speichern AWS Systems Manager. Weitere Informationen finden Sie unter [Was ist AWS Systems Manager?](#) im AWS Systems Manager-Benutzerhandbuch.

Sie können verwenden AWS SCT, um die Werte und Namen Ihrer Parameter automatisch zu aktualisieren. Aufgrund der Architekturunterschiede zwischen Apache Oozie und AWS Step Functions müssen Sie möglicherweise Ihre Parameter konfigurieren. AWS SCT kann einen bestimmten Parameternamen oder Wert in Ihren Quelldateien finden und durch neue Werte ersetzen. Weitere Informationen finden Sie unter [Schritt 3: Parameter konfigurieren](#).

Die folgende Abbildung zeigt das Architekturdiagramm der Apache Oozie-Konvertierung AWS Step Functions in.



Um die Konvertierung zu starten, erstellen Sie Ihr AWS SCT CLI-Skript und führen Sie es aus. Dieses Skript enthält den kompletten Befehlssatz zum Ausführen der Konvertierung. Sie können eine Vorlage des Apache Oozie Konvertierungsskripts herunterladen und bearbeiten. Weitere Informationen finden Sie unter [CLI-Szenarien abrufen](#).

Stellen Sie sicher, dass Ihr Skript die folgenden Schritte enthält.

Schritt 1: Stellen Sie eine Verbindung zu Ihren Quell- und Zieldiensten her

Erstellen Sie ein neues AWS SCT Projekt, um mit der Konvertierung Ihres Apache Oozie-Clusters zu beginnen. Stellen Sie als Nächstes eine Verbindung zu Ihren Quell- und Zieldiensten her. Stellen Sie sicher, dass Sie Ihre AWS Zielressourcen erstellen und bereitstellen, bevor Sie mit der Migration beginnen. Weitere Informationen finden Sie unter [Voraussetzungen für die Verwendung von Apache Oozie als Quelle](#).

In diesem Schritt verwenden Sie die folgenden AWS SCT CLI-Befehle.

- `CreateProject`— um ein neues AWS SCT Projekt zu erstellen.
- `AddSource`— um Ihre Apache Oozie-Quelldateien zu Ihrem AWS SCT Projekt hinzuzufügen.
- `ConnectSource`— um eine Verbindung zu Apache Oozie als Quelle herzustellen.
- `AddTarget`— um es Ihrem Projekt AWS Step Functions als Migrationsziel hinzuzufügen.

- `ConnectTarget`— um eine Verbindung herzustellenAWS Step Functions.

Beispiele für die Verwendung dieser AWS SCT CLI-Befehle finden Sie unter [Apache Oozie als Quelle verwenden](#).

Wenn Sie die `ConnectTarget` Befehle `ConnectSource` oder ausführen, wird AWS SCT versucht, die Verbindung zu Ihren Diensten herzustellen. Wenn der Verbindungsversuch fehlschlägt, wird die Ausführung der Befehle aus Ihrem CLI-Skript AWS SCT beendet und eine Fehlermeldung angezeigt.

Schritt 2: Richten Sie die Zuordnungsregeln ein

Nachdem Sie eine Verbindung zu Ihren Quell- und Zieldiensten hergestellt haben, richten Sie die Zuordnungsregeln ein. Eine Zuordnungsregel definiert das Migrationsziel für Ihre Apache Oozie-Quellworkflows und -Parameter. Weitere Hinweise zu Zuordnungsregeln finden Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#).

Verwenden Sie den `AddServerMapping` Befehl, um Quell- und Zielobjekte für die Konvertierung zu definieren. Dieser Befehl verwendet zwei Parameter: `sourceTreePath` und `targetTreePath`. Die Werte dieser Parameter beinhalten einen expliziten Pfad zu Ihren Quell- und Zielobjekten. Damit Apache Oozie AWS Step Functions konvertiert werden kann, müssen diese Parameter mit ETL beginnen.

Im folgenden Codebeispiel wird eine Zuordnungsregel für `AWS_STEP_FUNCTIONS` Objekte `OOZIE` und erstellt. Sie haben diese Objekte mit den `AddTarget` Befehlen `AddSource` und im vorherigen Schritt zu Ihrem AWS SCT Projekt hinzugefügt.

```
AddServerMapping
  -sourceTreePath: 'ETL.APACHE_OOZIE'
  -targetTreePath: 'ETL.AWS_STEP_FUNCTIONS'
/
```

Weitere Informationen zu dem `AddServerMapping` Befehl finden Sie in der [AWS Schema Conversion ToolCLI-Referenz](#).

Schritt 3: Parameter konfigurieren

Wenn Ihre Apache Oozie-Quellworkflows Parameter verwenden, müssen Sie möglicherweise deren Werte nach der Konvertierung in ändern. AWS Step Functions Möglicherweise müssen Sie auch neue Parameter hinzufügen, um sie mit Ihrem zu verwendenAWS Step Functions.

In diesem Schritt verwenden Sie die `AddTargetParameter` Befehle `AddParameterMapping` und

Verwenden Sie den `AddParameterMapping` Befehl, um die Parameterwerte in Ihren Quelldateien zu ersetzen. AWS SCT scannt Ihre Quelldateien, findet die Parameter nach Namen oder Wert und ändert ihre Werte. Sie können einen einzigen Befehl ausführen, um alle Ihre Quelldateien zu scannen. Sie definieren den Umfang der zu scannenden Dateien mithilfe eines der ersten drei Parameter aus der folgenden Liste. Dieser Befehl verwendet bis zu sechs Parameter.

- `filterName`— der Name des Filters für Ihre Quellobjekte. Mit dem `CreateFilter` Befehl können Sie einen Filter erstellen.
- `treePath`— der explizite Pfad zu Ihren Quellobjekten.
- `namePath`— der explizite Pfad zu einem bestimmten Quellobjekt.
- `sourceParameterName`— der Name Ihres Quellparameters.
- `sourceValue`— der Wert Ihres Quellparameters.
- `targetValue`— der Wert Ihres Zielparameters.

Das folgende Codebeispiel ersetzt alle Parameter, bei denen der Wert gleich ist, `c:\oozie\hive.py` durch den `s3://bucket-oozie/hive.py` Wert.

```
AddParameterMapping
-treePath: 'ETL.OOZIE.Applications'
-sourceValue: 'c:\oozie\hive.py'
-targetValue: 's3://bucket-oozie/hive.py'
/
```

Das folgende Codebeispiel ersetzt alle Parameter, bei denen der Name gleich ist, `nameNode` durch den `hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020` Wert.

```
AddParameterMapping
-treePath: 'ETL.OOZIE_SOURCE.Applications'
-sourceParameter: 'nameNode'
-targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'
/
```

Das folgende Codebeispiel ersetzt alle Parameter, bei denen der Name `nameNode` und der Wert gleich ist, `hdfs://ip-55.eu-west-1.compute.internal:8020` durch den Wert des `targetValue` Parameters.

```
AddParameterMapping
  -treePath: 'ETL.00ZIE_SOURCE.Applications'
  -sourceParameter: 'nameNode'
  -sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'
  -targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'
/
```

Verwenden Sie den `AddTargetParameter` Befehl, um Ihren Zieldateien zusätzlich zu einem vorhandenen Parameter aus Ihren Quelldateien einen neuen Parameter hinzuzufügen. Dieser Befehl verwendet dieselben Parameter wie der `AddParameterMapping` Befehl.

Im folgenden Codebeispiel `clusterId` wird der Zielparameter anstelle des `nameNode` Parameters hinzugefügt.

```
AddTargetParameter
  -treePath: 'ETL.00ZIE_SOURCE.Applications'
  -sourceParameter: 'nameNode'
  -sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'
  -targetParameter: 'clusterId'
  -targetValue: '1234567890abcdef0'
/
```

Weitere Informationen zu den `CreateFilter` Befehlen

`AddServerMapping`, `AddParameterMapping`, `AddTargetParameter`, und finden Sie in der [AWS Schema Conversion Tool CLI-Referenz](#).

Schritt 4: Erstellen Sie einen Bewertungsbericht

Bevor Sie mit der Konvertierung beginnen, empfehlen wir, einen Bewertungsbericht zu erstellen. Dieser Bericht fasst alle Migrationsaufgaben zusammen und beschreibt detailliert die Aktionspunkte, die während der Migration anfallen werden. Um sicherzustellen, dass Ihre Migration nicht fehlschlägt, sehen Sie sich diesen Bericht an und gehen Sie vor der Migration auf die Aktionspunkte ein. Weitere Informationen finden Sie unter [Berichte über die Migrationsbewertung](#).

In diesem Schritt verwenden Sie den `CreateReport` Befehl. Dieser Befehl verwendet zwei Parameter. Der erste Parameter beschreibt die Quellobjekte, für die ein Bewertungsbericht AWS SCT erstellt wird. Verwenden Sie dazu einen der folgenden Parameter: `filterName`, `treePath`, oder `namePath`. Dieser Parameter ist obligatorisch. Sie können auch einen optionalen booleschen Parameter hinzufügen. `forceLoad` Wenn Sie diesen Parameter auf `setzentru`, AWS SCT werden

automatisch alle untergeordneten Objekte für das Quellobjekt geladen, das Sie im `CreateReport` Befehl angeben.

Im folgenden Codebeispiel wird ein Bewertungsbericht für den `Applications` Knoten Ihrer Oozie-Quelldateien erstellt.

```
CreateReport
  -treePath: 'ETL.APACHE_00ZIE.Applications'
/
```

Anschließend können Sie eine Kopie des Bewertungsberichts als PDF oder als Datei mit kommagetrennten Werten (CSV) speichern. Verwenden Sie dazu den `SaveReportCSV` Befehl `SaveReportPDF` oder.

Der `SaveReportPDF` Befehl speichert eine Kopie Ihres Bewertungsberichts als PDF-Datei. Dieser Befehl verwendet vier Parameter. Der `file` Parameter ist obligatorisch, andere Parameter sind optional.

- `file`— der Pfad zur PDF-Datei und ihr Name.
- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Umfang Ihrer zu migrierenden Quellobjekte zu definieren.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `namePath`— der Pfad, der nur die Namen Ihrer Zielobjekte enthält, für die Sie eine Kopie des Bewertungsberichts speichern.

Der `SaveReportCSV` Befehl speichert Ihren Bewertungsbericht in CSV-Dateien. Dieser Befehl verwendet vier Parameter. Der `directory` Parameter ist obligatorisch, andere Parameter sind optional.

- `directory`— der Pfad zu dem Ordner, in dem die CSV-Dateien AWS SCT gespeichert sind.
- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Umfang Ihrer zu migrierenden Quellobjekte zu definieren.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `namePath`— der Pfad, der nur die Namen Ihrer Zielobjekte enthält, für die Sie eine Kopie des Bewertungsberichts speichern.

Im folgenden Codebeispiel wird eine Kopie des Bewertungsberichts in der `c:\sct\ar.pdf` Datei gespeichert.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

Im folgenden Codebeispiel wird eine Kopie des Bewertungsberichts als CSV-Datei im `c:\sct` Ordner gespeichert.

```
SaveReportCSV
-file:'c:\sct'
/
```

Weitere Informationen zu den `CreateReport` `SaveReportCSV` Befehlen `SaveReportPDF` und finden Sie in der [AWS Schema Conversion Tool CLI-Referenz](#).

Schritt 5: Konvertieren Sie Ihre Apache Oozie-Workflows auf mit AWS Step Functions AWS SCT

Nachdem Sie Ihr AWS SCT Projekt konfiguriert haben, konvertieren Sie Ihren Quellcode und wenden Sie ihn auf das an AWS Cloud.

In diesem Schritt verwenden Sie die `ApplyToTarget` Befehle `Convert` `SaveOnS3ConfigureStateMachine`, und.

Der `Migrate` Befehl migriert Ihre Quellobjekte zum Zielcluster. Dieser Befehl verwendet vier Parameter. Stellen Sie sicher, dass Sie den `treePath` Parameter `filter` oder angeben. Andere Parameter sind optional.

- `filter`— der Name des Filters, den Sie zuvor erstellt haben, um den Umfang Ihrer zu migrierenden Quellobjekte zu definieren.
- `namePath`— der explizite Pfad zu einem bestimmten Quellobjekt.
- `treePath`— der explizite Pfad zu Ihren Quelldatenbankobjekten, für die Sie eine Kopie des Bewertungsberichts speichern.
- `forceLoad`— Wenn auf `gesetztrue`, AWS SCT werden Datenbank-Metadatenbäume während der Migration automatisch geladen. Der Standardwert ist `false`.

Das folgende Codebeispiel konvertiert Dateien aus dem Applications Ordner in Ihren Oozie-Quelldateien.

```
Convert
  -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

Der SaveOnS3 lädt die Definitionen der State Machines in Ihren Amazon S3-Bucket hoch. Dieser Befehl verwendet den treePath Parameter. Um diesen Befehl auszuführen, verwenden Sie den Zielordner mit den Definitionen von State Machines als Wert dieses Parameters.

Im Folgenden wird der State machine definitions Ordner Ihres AWS_STEP_FUNCTIONS Zielobjekts in den Amazon S3-Bucket hochgeladen. AWS SCT verwendet den Amazon S3-Bucket, den Sie im [Voraussetzungen](#) Schritt im AWS Serviceprofil gespeichert haben.

```
SaveOnS3
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
/
```

Der ConfigureStateMachine Befehl konfiguriert Zustandsmaschinen. Dieser Befehl verwendet bis zu sechs Parameter. Stellen Sie sicher, dass Sie den Zielbereich mithilfe eines der ersten drei Parameter aus der folgenden Liste definieren.

- `filterName`— der Name des Filters für Ihre Zielobjekte. Mit dem `CreateFilter` Befehl können Sie einen Filter erstellen.
- `treePath`— der explizite Pfad zu Ihren Zielobjekten.
- `namePath`— der explizite Pfad zu einem bestimmten Zielobjekt.
- `iamRole`— der Amazon-Ressourcenname (ARN) der IAM-Rolle, die Zugriff auf Ihre Step-Machines bietet. Dieser Parameter muss angegeben werden.

Im folgenden Codebeispiel werden State-Maschinen konfiguriert, die AWS_STEP_FUNCTIONS mithilfe der *IAM-Rolle* `role_name` definiert sind.

```
ConfigureStateMachine
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
  -role: 'arn:aws:iam::555555555555:role/role_name'
/
```


Der `ApplyToTarget` Befehl wendet Ihren konvertierten Code auf den Zielservers an. Verwenden Sie einen der folgenden Parameter, um diesen Befehl auszuführen: `filterName`, `treePath`, oder `namePath` um die anzuwendenden Zielobjekte zu definieren.

Im folgenden Codebeispiel wird die `app_wp` State-Maschine auf angewendet AWS Step Functions.

```
ApplyToTarget
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machines.app_wp'
/
```

Um sicherzustellen, dass Ihr konvertierter Code dieselben Ergebnisse wie Ihr Quellcode liefert, können Sie das AWS SCT Erweiterungspaket verwenden. Dies ist eine Reihe von AWS Lambda Funktionen, die Ihre Apache Oozie-Funktionen emulieren, die AWS Step Functions nicht unterstützt werden. Um dieses Erweiterungspaket zu installieren, können Sie den `CreateLambdaExtPack` Befehl verwenden.

Dieser Befehl verwendet bis zu fünf Parameter. Stellen Sie sicher, dass Sie **Oozie2SF** für `extPackId`. In diesem Fall AWS SCT erstellt es ein Erweiterungspaket für die Quellfunktionen von Apache Oozie.

- `extPackId`— die eindeutige Kennung für eine Reihe von Lambda-Funktionen. Dieser Parameter muss angegeben werden.
- `tempDirectory`— der Pfad, in dem temporäre Dateien gespeichert werden AWS SCT können. Dieser Parameter muss angegeben werden.
- `awsProfile`— der Name Ihres AWS Profils.
- `lambdaExecRoles`— die Liste der Amazon Resource Names (ARNs) der Ausführungsrollen, die für Lambda-Funktionen verwendet werden sollen.
- `createInvokeRoleFlag`— das boolesche Flag, das angibt, ob eine Ausführungsrolle für erstellt werden soll. AWS Step Functions

Um das Erweiterungspaket zu installieren und zu verwenden, stellen Sie sicher, dass Sie die erforderlichen Berechtigungen bereitstellen. Weitere Informationen finden Sie unter [Berechtigungen für die Verwendung von AWS Lambda Funktionen im Erweiterungspaket](#).

Weitere Informationen zu den `CreateLambdaExtPack` Befehlen

`ConvertSaveOnS3`, `ConfigureStateMachine`, `ApplyToTarget`, und finden Sie in der [AWS Schema Conversion Tool CLI-Referenz](#).

Ihr CLI-Skript ausführen

Nachdem Sie die Bearbeitung Ihres AWS SCT CLI-Skripts abgeschlossen haben, speichern Sie es als Datei mit der `.scts` Erweiterung. Jetzt können Sie Ihr Skript aus dem `app` Ordner Ihres AWS SCT Installationspfads ausführen. Führen Sie dazu den folgenden Befehl aus.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\oozie.scts"
```

Ersetzen Sie im vorherigen Beispiel *script_path* durch den Pfad zu Ihrer Datei mit dem CLI-Skript. Weitere Hinweise zum Ausführen von CLI-Skripten in finden Sie AWS SCT unter [Skriptmodus](#).

Apache Oozie-Knoten, die konvertiert AWS SCT werden können in AWS Step Functions

Sie können AWS SCT Apache Oozie Action-Knoten und Control-Flow-Knoten in AWS Step Functions konvertieren.

Zu den unterstützten Aktionsknoten gehören die folgenden:

- Aktion im Bienenstock
- Hive2-Aktion
- Funke Action
- MapReduceStreaming-Aktion
- Java-Aktion
- DistCp-Aktion
- Aktion Schwein
- Sqoop-Aktion
- FS-Aktion
- Shell-Aktion

Zu den unterstützten Kontrollflussknoten gehören die folgenden:

- Aktion starten
- Aktion beenden
- Aktion beenden

- Entscheidung, Aktion
- Gabelwirkung
- Nimm an der Aktion teil

Verwenden von AWS SCT mit AWS DMS

Verwendung einer AWS SCT Replikationsagenten mit AWS DMS

Für sehr große Datenbankmigrationen können Sie eine AWS SCT Replikationsagent (`aws-schema-conversion-tool-dms-agent`) um Daten aus Ihrer lokalen Datenbank in Amazon S3 oder eine AWS SnowballEdge-Gerät. Der Replikationsagent kann in Verbindung mit AWS DMS eingesetzt und auch im Hintergrund ausgeführt werden, während AWS SCT geschlossen ist.

Bei der Arbeit mit AWS SnowballEdge, der AWS SCT Agent repliziert Daten auf die AWS Snowball-Gerät. Das Gerät wird dann an gesendet AWS und die Daten in einen Amazon S3 S3-Bucket geladen. Während dieser Zeit wird der AWS SCT-Agent weiterhin ausgeführt. Der Agent nimmt dann die Daten aus Amazon S3 und kopiert diese in den Zielpunkt.

Weitere Informationen finden Sie unter [Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift](#).

Verwendung einer AWS SCT Datenextraktionsagent mit AWS DMS

In :AWS SCT finden Sie einen Datenextraktionsagenten (`aws-schema-conversion-tool-extractor`) das hilft, die Migrationen von Apache Cassandra auf Amazon DynamoDB zu vereinfachen. Cassandra und DynamoDB sind NoSQL-Datenbanken, unterscheiden sich jedoch in Systemarchitektur und Datendarstellung. Sie können assistentenbasierte Workflows in AWS SCT um den Migrationsprozess von Cassandra-zu-Dynamodb zu automatisieren. AWS SCT ist in integriert. AWS Database Migration Service (AWS DMS) um die eigentliche Migration durchzuführen.

Weitere Informationen finden Sie unter [Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift](#).

Erhöhung der Protokollierungsstufen bei Verwendung AWS SCT mit AWS DMS

Sie können die Protokollierungsstufen erhöhen, wenn Sie AWS SCT mit AWS DMS, zum Beispiel wenn Sie mit arbeiten müssen AWS Support.

Nach Installation von AWS SCT und den erforderlichen Treibern öffnen Sie die Anwendung, indem Sie die AWS SCT. Wenn eine Update-Benachrichtigung angezeigt wird, können Sie vor oder nach

Abschluss Ihres Projekts aktualisieren. Wenn ein automatisches Projektfenster geöffnet wird, schließen Sie das Fenster und erstellen Sie manuell ein Projekt.

So erhöhen Sie die Protokollierungsstufen bei Verwendung AWS SCT mit AWS DMS

1. Auf der **Einstellungen**-Auf, wählen Sie **Globale Einstellungen** aus.
2. In der **Globale Einstellungen**-Auf, wählen Sie **Protokollierung** aus.
3. Für **Debug-Modus**, wählen Sie **Wahr** aus.
4. Aus der **Nachrichten-Level** können Sie die folgenden Arten von Protokollen ändern:
 - Allgemeines
 - Loader
 - Parser
 - printer
 - Resolver
 - Telemetrie
 - Converter

Standardmäßig sind alle Nachrichtenstufen auf **Informationen** aus.

5. Wählen Sie eine Protokollierungsstufe für alle Nachrichtentypen aus, die Sie ändern möchten:
 - Trace (detaillierteste Protokollierung)
 - Debuggen
 - Informationen
 - Warnung
 - Fehler (am wenigsten detaillierte Protokollierung)
 - Critical
 - zwingend erforderlich
6. Klicken Sie auf **Anwenden** um die Einstellungen für Ihr Projekt zu ändern.
7. Klicken Sie auf **OKAY** Schließen Sie den **Globale Einstellungen**-Fenster.

Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift

Sie können einen AWS SCT Agenten verwenden, um Daten aus Ihrem lokalen Data Warehouse zu extrahieren und zu Amazon Redshift zu migrieren. Der Agent extrahiert Ihre Daten und lädt die Daten entweder auf Amazon S3 oder, bei groß angelegten Migrationen, auf ein AWS Snowball Edge-Gerät hoch. Anschließend können Sie einen AWS SCT Agenten verwenden, um die Daten nach Amazon Redshift zu kopieren.

Alternativ können Sie AWS Database Migration Service (AWS DMS) verwenden, um Daten zu Amazon Redshift zu migrieren. Der Vorteil von AWS DMS ist die Unterstützung der laufenden Replikation (Erfassung von Änderungsdaten). Um die Geschwindigkeit der Datenmigration zu erhöhen, sollten Sie jedoch mehrere AWS SCT Agenten parallel verwenden. Unseren Tests zufolge migrieren AWS SCT Agenten Daten schneller als AWS DMS um 15 bis 35 Prozent. Der Geschwindigkeitsunterschied ist auf die Datenkomprimierung, die Unterstützung der parallel Migration von Tabellenpartitionen und unterschiedliche Konfigurationseinstellungen zurückzuführen. Weitere Informationen finden Sie unter [Verwenden einer Amazon-Redshift-Datenbank als Ziel für AWS Database Migration Service](#).

Amazon S3 ist ein Speicher- und Abrufservice. Um Objekte in Amazon S3 zu speichern, laden Sie die Datei hoch, die Sie in einem Amazon S3-Bucket speichern möchten. Beim Hochladen der Datei können Sie Berechtigungen für das Objekt sowie beliebige Metadaten festlegen.

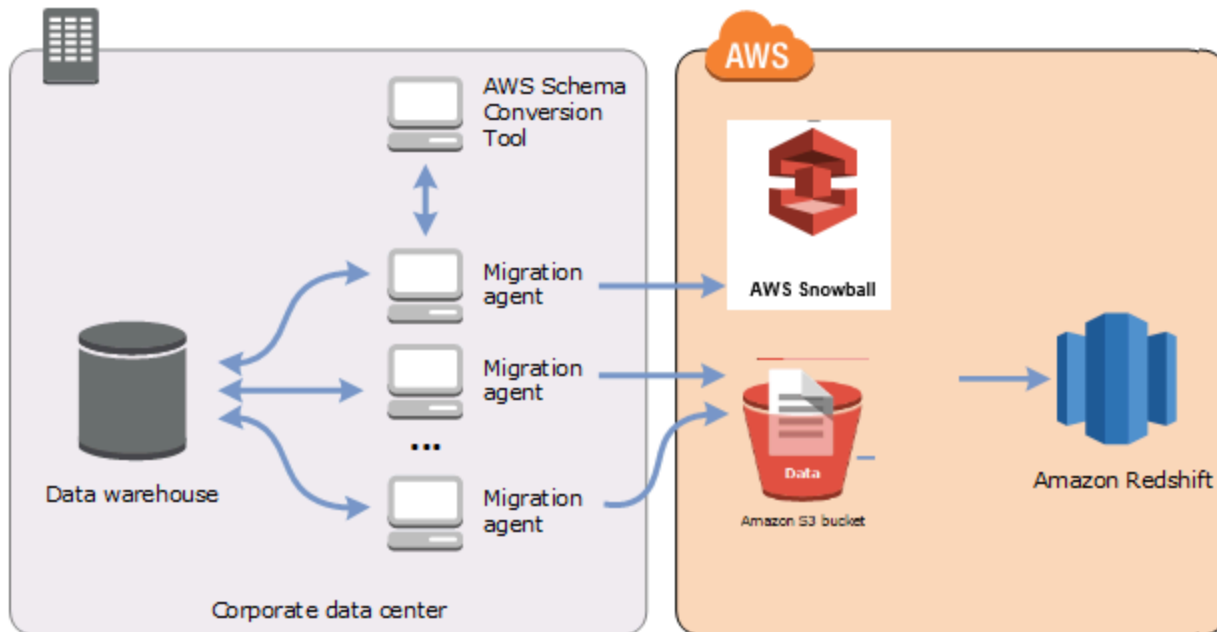
Large-scale migrations (Umfangreiche Migrationen)

Umfangreiche Datenmigrationen können viele Terabyte an Informationen umfassen und können durch die Netzwerkleistung und die schiere Menge an Daten, die verschoben werden müssen, gebremst werden. AWS Snowball Edge ist ein AWS Dienst, mit dem Sie Daten mithilfe einer eigenen Appliance mit faster-than-network hoher Geschwindigkeit in die Cloud übertragen können. Ein AWS Snowball Edge-Gerät kann bis zu 100 TB an Daten aufnehmen. Es verwendet eine 256-Bit-Verschlüsselung und ein branchenübliches Trusted Platform Module (TPM), um sowohl die Sicherheit als auch die Vollständigkeit Ihrer Daten zu gewährleisten. chain-of-custody AWS SCT funktioniert mit Edge-Geräten. AWS Snowball

Wenn Sie ein AWS Snowball Edge-Gerät verwenden AWS SCT , migrieren Sie Ihre Daten in zwei Schritten. Zunächst verarbeiten Sie AWS SCT die Daten lokal und verschieben diese Daten dann auf das AWS Snowball Edge-Gerät. Anschließend senden Sie das Gerät AWS mithilfe des AWS

Snowball Edge-Prozesses an und laden die Daten dann AWS automatisch in einen Amazon S3 S3-Bucket. Wenn die Daten auf Amazon S3 verfügbar sind, migrieren Sie AWS SCT sie anschließend zu Amazon Redshift. Datenextraktionsagenten können im Hintergrund arbeiten, während der Vorgang geschlossen AWS SCT ist.

Das folgende Diagramm veranschaulicht das unterstützte Szenario.



Agenten für die Datenextraktion werden derzeit für die folgenden Quell-Data Warehouses unterstützt:

- Azure Synapse Analytics
- BigQuery
- Greenplum-Datenbank (Version 4.3)
- Microsoft SQL Server (Version 2008 und höher)
- Netezza (Version 7.0.3 und höher)
- Oracle (Version 10 und höher)
- Schneeflocke (Version 3)
- Teradata (Version 13 und höher)
- Vertica (Version 7.2.2 und höher)

Sie können eine Verbindung zu FIPS-Endpunkten für Amazon Redshift herstellen, wenn Sie die Sicherheitsanforderungen des Federal Information Processing Standard (FIPS) erfüllen müssen. FIPS-Endpunkte sind in den folgenden Regionen verfügbar: AWS

- Region USA Ost (Nord-Virginia) (redshift-fips.us-east-1.amazonaws.com)
- Region USA Ost (Ohio) (redshift-fips.us-east-2.amazonaws.com)
- Region USA West (Nordkalifornien) (redshift-fips.us-west-1.amazonaws.com)
- Region USA West (Oregon) (redshift-fips.us-west-2.amazonaws.com)

Verwenden Sie die Informationen in den folgenden Themen, um zu erfahren, wie Sie mit Agenten für die Datenextraktion arbeiten.

Themen

- [Voraussetzungen für die Verwendung von Datenextraktionsagenten](#)
- [Installation von Extraktionsagenten](#)
- [Konfiguration von Extraktionsagenten](#)
- [Registrierung von Extraktionsagenten mit dem AWS Schema Conversion Tool](#)
- [Informationen für einen AWS SCT Agenten ausblenden und wiederherstellen](#)
- [Regeln für die Datenmigration erstellen in AWS SCT](#)
- [Ändern der Extraktor- und Kopiereinstellungen aus den Projekteinstellungen](#)
- [Sortieren von Daten vor der Migration mit AWS SCT](#)
- [Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen](#)
- [Exportieren und Importieren einer AWS SCT Datenextraktionsaufgabe](#)
- [Datenextraktion mit einem AWS Snowball Edge-Gerät](#)
- [Ausgabe der Aufgabe zur Datenextraktion](#)
- [Verwenden Sie virtuelle Partitionierung mit AWS Schema Conversion Tool](#)
- [Native Partitionierung verwenden](#)
- [Migration von LOBs zu Amazon Redshift](#)
- [Bewährte Methoden und Problemlösung für Datenextraktionsagenten](#)

Voraussetzungen für die Verwendung von Datenextraktionsagenten

Bevor Sie mit Datenextraktionsagenten arbeiten, fügen Sie Ihrem Amazon Redshift-Benutzer die erforderlichen Berechtigungen für Amazon Redshift als Ziel hinzu. Weitere Informationen finden Sie unter [Berechtigungen für Amazon Redshift als Ziel](#).

Speichern Sie anschließend Ihre Amazon S3 S3-Bucket-Informationen und richten Sie Ihren Secure Sockets Layer (SSL) Trust and Key Store ein.

Amazon S3 S3-Einstellungen

Nachdem Ihre Agenten Ihre Daten extrahiert haben, laden sie sie in Ihren Amazon S3 S3-Bucket hoch. Bevor Sie fortfahren, müssen Sie die Anmeldeinformationen angeben, um eine Verbindung zu Ihrem AWS Konto und Ihrem Amazon S3 S3-Bucket herzustellen. Sie speichern Ihre Anmeldeinformationen und Bucket-Informationen in einem Profil in den globalen Anwendungseinstellungen und verknüpfen das Profil dann mit Ihrem AWS SCT Projekt. Wählen Sie bei Bedarf Allgemeine Einstellungen aus, um ein neues Profil zu erstellen. Weitere Informationen finden Sie unter [Speichern von AWS Serviceprofilen in AWS SCT](#).

Um Daten in Ihre Amazon Redshift Redshift-Zieldatenbank zu migrieren, benötigt der AWS SCT Datenextraktionsagent die Erlaubnis, in Ihrem Namen auf den Amazon S3 S3-Bucket zuzugreifen. Um diese Berechtigung zu erteilen, erstellen Sie einen AWS Identity and Access Management (IAM-) Benutzer mit der folgenden Richtlinie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectTagging",
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::bucket_name/*",
        "arn:aws:s3:::bucket_name"
      ],
    }
  ],
}
```

```
    "Effect": "Allow"
  },
  {
    "Action": [
      "s3:ListBucket",
      "s3:GetBucketLocation"
    ],
    "Resource": [
      "arn:aws:s3:::bucket_name"
    ],
    "Effect": "Allow"
  },
  {
    "Effect": "Allow",
    "Action": "s3:ListAllMyBuckets",
    "Resource": "*"
  },
  {
    "Action": [
      "iam:GetUser"
    ],
    "Resource": [
      "arn:aws:iam::111122223333:user/DataExtractionAgentName"
    ],
    "Effect": "Allow"
  }
]
}
```

Ersetzen Sie es im vorherigen Beispiel *bucket_name* durch den Namen Ihres Amazon S3 S3-Buckets. Ersetzen Sie es dann *111122223333:user/DataExtractionAgentName* durch den Namen Ihres IAM-Benutzers.

IAM-Rollen annehmen

Für zusätzliche Sicherheit können Sie AWS Identity and Access Management (IAM) -Rollen für den Zugriff auf Ihren Amazon S3 S3-Bucket verwenden. Erstellen Sie dazu einen IAM-Benutzer für Ihre Datenextraktionsagenten ohne jegliche Berechtigungen. Erstellen Sie anschließend eine IAM-Rolle, die den Zugriff auf Amazon S3 ermöglicht, und geben Sie die Liste der Dienste und Benutzer an, die diese Rolle übernehmen können. Weitere Informationen finden Sie unter [IAM-Rollen](#) im IAM-Benutzerhandbuch.

So konfigurieren Sie IAM-Rollen für den Zugriff auf Ihren Amazon S3 S3-Bucket

1. Erstellen Sie einen neuen IAM-Benutzer. Wählen Sie für Benutzeranmeldedaten den Zugriffstyp `Programmatic` aus.
2. Konfigurieren Sie die Hostumgebung so, dass Ihr Datenextraktionsagent die dafür vorgesehene Rolle übernehmen kann AWS SCT . Stellen Sie sicher, dass der Benutzer, den Sie im vorherigen Schritt konfiguriert haben, den Datenextraktionsagenten die Nutzung der Anmeldeinformationsanbieterkette ermöglicht. Weitere Informationen finden Sie unter [Verwenden von Anmeldeinformationen](#) im AWS SDK for Java Entwicklerhandbuch.
3. Erstellen Sie eine neue IAM-Rolle, die Zugriff auf Ihren Amazon S3 S3-Bucket hat.
4. Ändern Sie den Vertrauensbereich dieser Rolle so, dass der Benutzer, den Sie zuvor erstellt haben, die Rolle übernimmt. Im folgenden Beispiel `111122223333:user/DataExtractionAgentName` ersetzen Sie ihn durch den Namen Ihres Benutzers.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/DataExtractionAgentName"
  },
  "Action": "sts:AssumeRole"
}
```

5. Ändern Sie den Vertrauensbereich dieser Rolle in `Vertrauenredshift.amazonaws.com`, um die Rolle zu übernehmen.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "redshift.amazonaws.com"
    ]
  },
  "Action": "sts:AssumeRole"
}
```

6. Ordnen Sie diese Rolle Ihrem Amazon Redshift Redshift-Cluster zu.

Jetzt können Sie Ihren Datenextraktionsagenten in AWS SCT ausführen.

Wenn Sie IAM-Rollenannahme verwenden, funktioniert die Datenmigration wie folgt. Der Datenextraktionsagent wird gestartet und ruft die Benutzeranmeldeinformationen mithilfe der Anmeldeinformationsanbieterkette ab. Als Nächstes erstellen Sie eine Datenmigrationsaufgabe in AWS SCT, geben dann die IAM-Rolle an, die die Datenextraktionsagenten übernehmen sollen, und starten die Aufgabe. AWS Security Token Service (AWS STS) generiert temporäre Anmeldeinformationen für den Zugriff auf Amazon S3. Der Datenextraktionsagent verwendet diese Anmeldeinformationen, um Daten auf Amazon S3 hochzuladen.

AWS SCT stellt dann Amazon Redshift die IAM-Rolle zur Verfügung. Im Gegenzug erhält Amazon Redshift neue temporäre Anmeldeinformationen für AWS STS den Zugriff auf Amazon S3. Amazon Redshift verwendet diese Anmeldeinformationen, um Daten von Amazon S3 in Ihre Amazon Redshift Redshift-Tabelle zu kopieren.

Sicherheitseinstellungen

Die AWS Schema Conversion Tool und die Extraktionsagenten können über Secure Sockets Layer (SSL) kommunizieren. Richten Sie zum Aktivieren von SSL einen Vertrauensspeicher und einen Schlüsselspeicher ein.

So richten Sie die sichere Kommunikation mit dem Agenten für die Datenextraktion ein

1. Starten Sie den AWS Schema Conversion Tool.
2. Öffnen Sie das Einstellungsmenü und wählen Sie dann Allgemeine Einstellungen. Das Dialogfeld Global settings wird angezeigt.
3. Wählen Sie die Option Security (Sicherheit) aus.
4. Wählen Sie Vertrauens- und Schlüsselspeicher generieren oder Bestehenden Vertrauensspeicher auswählen.

Wenn Sie Vertrauens- und Schlüsselspeicher generieren wählen, geben Sie anschließend den Namen und das Passwort für den Vertrauens- und Schlüsselspeicher sowie den Pfad zum Speicherort für die generierten Dateien an. Diese Dateien werden in späteren Schritten verwendet.

Wenn Sie Bestehenden Vertrauensspeicher auswählen wählen, geben Sie anschließend das Passwort und den Dateinamen für die Vertrauens- und Schlüsselspeicher an. Diese Dateien werden in späteren Schritten verwendet.

5. Nachdem Sie den Vertrauensspeicher und den Schlüsselspeicher angegeben haben, wählen Sie OK, um das Dialogfeld Allgemeine Einstellungen zu schließen.

Konfiguration der Umgebung für Datenextraktionsagenten

Sie können mehrere Datenextraktionsagenten auf einem einzigen Host installieren. Wir empfehlen jedoch, einen Datenextraktionsagenten auf einem Host auszuführen.

Um Ihren Datenextraktionsagenten auszuführen, stellen Sie sicher, dass Sie einen Host mit mindestens vier vCPUs und 32 GB Arbeitsspeicher verwenden. Stellen Sie außerdem den verfügbaren Mindestspeicher AWS SCT auf mindestens vier GB ein. Weitere Informationen finden Sie unter [Konfigurieren zusätzlichen](#).

Die optimale Konfiguration und Anzahl der Agent-Hosts hängen von der spezifischen Situation jedes Kunden ab. Achten Sie darauf, Faktoren wie die Menge der zu migrierenden Daten, die Netzwerkbandbreite, die Zeit für das Extrahieren von Daten usw. zu berücksichtigen. Sie können zuerst einen Machbarkeitsnachweis (PoC) durchführen und dann Ihre Datenextraktionsagenten und Hosts entsprechend den Ergebnissen dieses PoC konfigurieren.

Installation von Extraktionsagenten

Wir empfehlen, mehrere Agenten für die Datenextraktion auf einzelnen Computern zu installieren. Dabei sollte auf dem Computer, auf dem das AWS Schema Conversion Tool ausgeführt wird, kein Agent installiert werden.

Agenten für die Datenextraktion werden derzeit auf folgenden Betriebssystemen installiert:

- Microsoft Windows
- Red Hat Enterprise Linux (RHEL) 6.0
- Ubuntu Linux (Version 14.04 und höher)

Gehen Sie wie folgt vor, um Agenten für die Datenextraktion zu installieren. Wiederholen Sie den Vorgang auf jedem Computer, auf dem Sie einen Agenten installieren möchten.

So installieren Sie einen Agenten für die Datenextraktion

1. Wenn Sie die AWS SCT Installationsdatei noch nicht heruntergeladen haben, folgen Sie den Anweisungen unter, [Installation, Überprüfung und Aktualisierung AWS SCT](#) um sie herunterzuladen. Die ZIP-Datei, die die Installationsdatei enthält, enthält auch die AWS SCT Installationsdatei für den Extraktionsagenten.

- Laden Sie die neueste Version von Amazon Corretto 11 herunter und installieren Sie sie. Weitere Informationen finden Sie unter [Downloads für Amazon Corretto 11](#) im Amazon Corretto 11-Benutzerhandbuch.
- Die Installationsdatei für den Agenten für die Datenextraktion befindet sich im Unterordner "Agents". Nachfolgend ist für jedes Betriebssystem die richtige Datei für die Installation des Agenten für die Datenextraktion gezeigt.

Betriebssystem	Dateiname
Microsoft Windows	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .msi
RHEL	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .x86_64.rpm
Ubuntu Linux	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .deb

- Installieren Sie den Extraktionsagenten auf einem separaten Computer, indem Sie die Installationsdatei auf den neuen Computer kopieren.
- Führen Sie die Installationsdatei aus. Befolgen Sie die Anleitung für Ihr Betriebssystem:

Betriebssystem	Installationsanleitungen
Microsoft Windows	Doppelklicken Sie auf die Datei, um das Installationsprogramm auszuführen.
RHEL	Führen Sie die folgenden Befehle in dem Ordner aus, in den Sie die Datei heruntergeladen oder verschoben haben. <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; margin-top: 10px;"> <pre>sudo rpm -ivh aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .x86_64.rpm sudo ./sct-extractor-setup.sh --config</pre> </div>
Ubuntu Linux	Führen Sie die folgenden Befehle in dem Ordner aus, in den Sie die Datei heruntergeladen oder verschoben haben.

Betriebssystem	Installationsanleitungen
	<pre>sudo dpkg -i aws-schema-conversion-tool-extractor -2.0.1. <i>build-number</i> .deb sudo ./sct-extractor-setup.sh --config</pre>

6. Wählen Sie Weiter, akzeptieren Sie die Lizenzvereinbarung und wählen Sie Weiter.
7. Geben Sie den Pfad zur Installation des AWS SCT Datenextraktionsagenten ein und wählen Sie Weiter.
8. Wählen Sie Installieren, um Ihren Datenextraktionsagenten zu installieren.

AWS SCT installiert Ihren Datenextraktionsagenten. Um die Installation abzuschließen, konfigurieren Sie Ihren Datenextraktionsagenten. AWS SCT startet automatisch das Konfigurations-Setup-Programm. Weitere Informationen finden Sie unter [Konfiguration von Extraktionsagenten](#).

9. Wählen Sie Fertig stellen, um den Installationsassistenten zu schließen, nachdem Sie Ihren Datenextraktionsagenten konfiguriert haben.

Konfiguration von Extraktionsagenten

Gehen Sie wie folgt vor, um Agenten für die Datenextraktion zu konfigurieren. Wiederholen Sie den Vorgang auf jedem Computer, auf dem Sie einen Agenten installiert haben.

So konfigurieren Sie einen Agenten für die Datenextraktion

1. Starten Sie das Konfigurations-Setup-Programm:
 - AWS SCT Startet das Konfigurations-Setup-Programm unter Windows automatisch während der Installation eines Datenextraktionsagenten.

Bei Bedarf können Sie das Setup-Programm manuell starten. Führen Sie dazu die `ConfigAgent.bat` Datei in Windows aus. Sie finden diese Datei in dem Ordner, in dem Sie den Agenten installiert haben.

- Führen Sie die `sct-extractor-setup.sh` Datei in RHEL und Ubuntu von dem Ort aus, an dem Sie den Agenten installiert haben.

Das Installationsprogramm fordert Sie zur Eingabe von Informationen auf. Es ist jeweils ein Standardwert vorkonfiguriert.

2. Akzeptieren Sie bei jeder Aufforderung den Standardwert, oder geben Sie einen neuen Wert ein.

Geben Sie folgende Informationen an:

- Geben Sie für Listening-Port die Portnummer ein, auf der der Agent lauscht.
- Geben Sie für Add a source vendor den Wert yes ein und geben Sie dann Ihre Data Warehouse-Quellplattform ein.
- Geben Sie für JDBC-Treiber den Speicherort ein, an dem Sie die JDBC-Treiber installiert haben.
- Geben Sie unter Arbeitsordner den Pfad ein, in dem der AWS SCT Datenextraktionsagent die extrahierten Daten speichern soll. Das Arbeitsverzeichnis kann sich auf einem anderen Computer als der Agent befinden. Ein einzelnes Arbeitsverzeichnis kann für mehrere Agenten auf verschiedenen Computern verwendet werden.
- Geben Sie für SSL-Kommunikation aktivieren den Wert yes ein.
- Geben Sie für Schlüsselspeicher den Speicherort der Schlüsselspeicherdatei ein.
- Geben Sie unter Schlüsselspeicher-Passwort das Passwort für den Schlüsselspeicher ein.
- Geben Sie für Client-SSL-Authentifizierung aktivieren den Wert yes ein.
- Geben Sie für Trust Store den Speicherort der Trust Store-Datei ein.
- Geben Sie unter Trust Store-Passwort das Passwort für den Trust Store ein.

Das Installationsprogramm aktualisiert die Einstellungsdatei für den Agenten für die Datenextraktion. Die Einstellungsdatei heißt `settings.properties` und befindet sich im Installationsverzeichnis des Agenten für die Datenextraktion.

Nachfolgend sehen Sie eine Beispielleinstellungsdatei.

```
$ cat settings.properties
#extractor.start.fetch.size=20000
#extractor.out.file.size=10485760
#extractor.source.connection.pool.size=20
#extractor.source.connection.pool.min.evictable.idle.time.millis=30000
#extractor.extracting.thread.pool.size=10
vendor=TERADATA
```

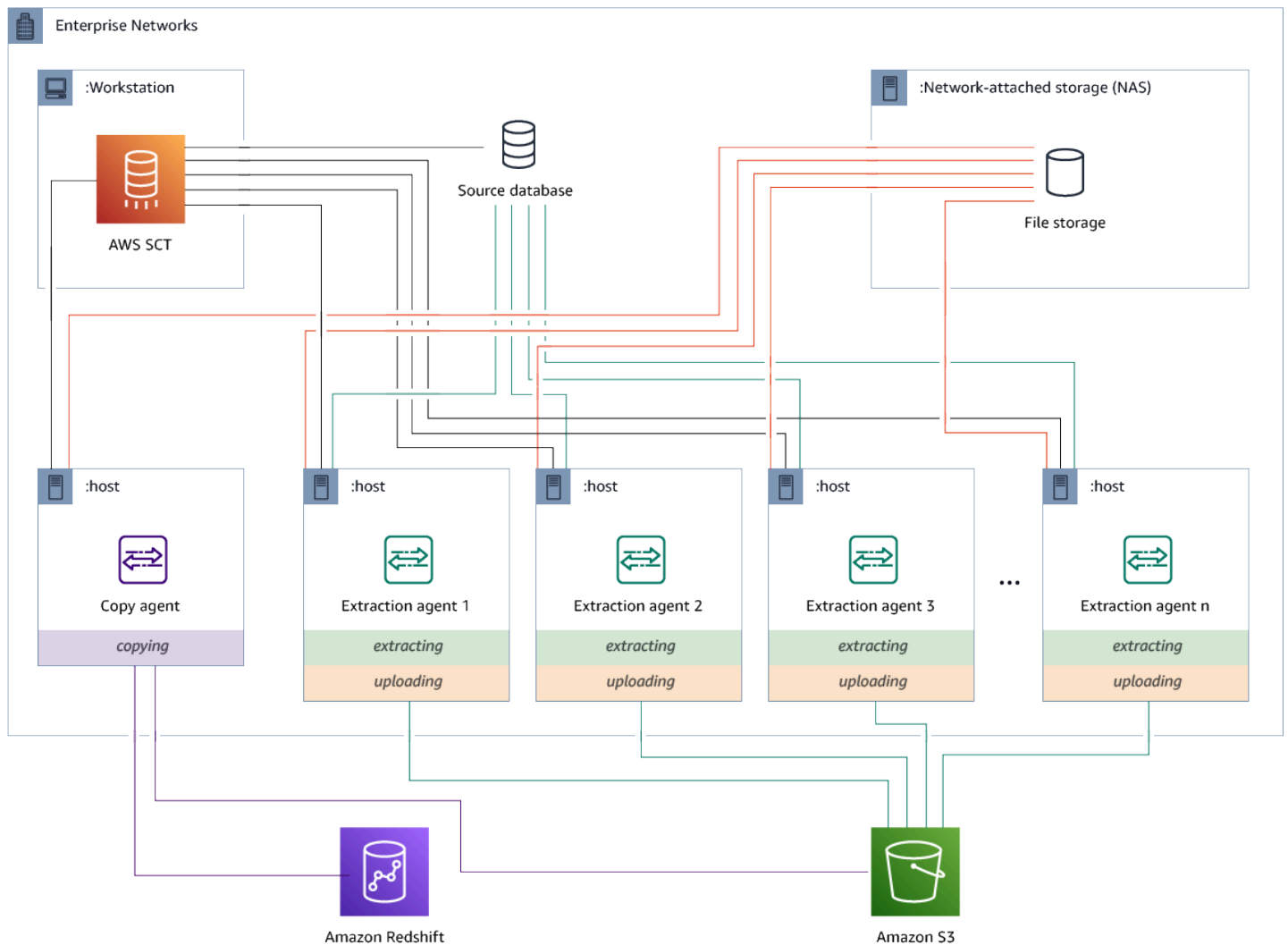


```
driver.jars=/usr/share/lib/jdbc/terajdbc4.jar
port=8192
redshift.driver.jars=/usr/share/lib/jdbc/RedshiftJDBC42-1.2.43.1067.jar
working.folder=/data/sct
extractor.private.folder=/home/ubuntu
ssl.option=OFF
```

Um die Konfigurationseinstellungen zu ändern, können Sie die `settings.properties` Datei mit einem Texteditor bearbeiten oder die Agentenkonfiguration erneut ausführen.

Installation und Konfiguration von Extraktionsagenten mit speziellen Kopieragenten

Sie können Extraktionsagenten in einer Konfiguration mit gemeinsam genutztem Speicher und einem dedizierten Kopieragenten installieren. Das folgende Diagramm veranschaulicht dieses Szenario.



Diese Konfiguration kann nützlich sein, wenn ein Quelldatenbankserver bis zu 120 Verbindungen unterstützt und Ihr Netzwerk über ausreichend Speicherplatz verfügt. Gehen Sie wie folgt vor, um Extraktionsagenten zu konfigurieren, die über einen eigenen Kopieragent verfügen.

So installieren und konfigurieren Sie Extraktionsagenten und einen speziellen Kopieragent

1. Stellen Sie sicher, dass das Arbeitsverzeichnis aller Extraktionsagenten denselben Ordner auf dem gemeinsam genutzten Speicher verwendet.
2. Installieren Sie die Extraktor-Agenten, indem Sie die Schritte unter befolgen. [Installation von Extraktionsagenten](#)
3. Konfigurieren Sie die Extraktionsagenten, indem Sie die Schritte unter befolgen [Konfiguration von Extraktionsagenten](#), aber geben Sie nur den Quell-JDBC-Treiber an.

4. Konfigurieren Sie einen dedizierten Kopieragenten, indem Sie die Schritte unter [Konfiguration von Extraktionsagenten](#) befolgen, aber nur einen Amazon Redshift JDBC-Treiber angeben.

Extraktionsagenten werden gestartet

Gehen Sie wie folgt vor, um Agenten für die Datenextraktion zu starten. Wiederholen Sie den Vorgang auf jedem Computer, auf dem Sie einen Agenten installiert haben.

Agenten für die Datenextraktion sind Listener. Wenn Sie einen Agenten auf diese Weise starten, wartet er auf Anweisungen. Sie senden dem Agenten später Anweisungen, Daten aus Ihrem Data Warehouse zu extrahieren.

So starten Sie einen Agenten für die Datenextraktion

- Führen Sie auf dem Computer, auf dem der Agent für die Datenextraktion installiert ist, den Befehl für Ihr Betriebssystem wie in der folgenden Tabelle dargestellt aus:

Betriebssystem	Befehl starten
Microsoft Windows	Doppelklicken Sie auf die Batch-Datei <code>StartAgent.bat</code> .
RHEL	Führen Sie in dem Verzeichnispfad, in dem Sie den Agenten installiert haben, den folgenden Befehl aus: <code>sudo initctl start sct-extractor</code>
Ubuntu Linux	Führen Sie in dem Verzeichnispfad, in dem Sie den Agenten installiert haben, den folgenden Befehl aus. Verwenden Sie den korrekten Befehl für Ihre Ubuntu-Version. Ubuntu 14.04: <code>sudo initctl start sct-extractor</code> Ubuntu 15.04 und höher: <code>sudo systemctl start sct-extractor</code>

Führen Sie zum Prüfen des Status des Agenten denselben Befehl aus und ersetzen Sie dabei `start` durch `status`.

Führen Sie zum Beenden des Agenten denselben Befehl aus und ersetzen Sie dabei `start` durch `stop`.

Registrierung von Extraktionsagenten mit dem AWS Schema Conversion Tool

Sie verwalten Ihre Extraktionsagenten mithilfe von AWS SCT. Die Agenten für die Datenextraktion sind Listener. Wenn sie Anweisungen von erhalten AWS SCT, extrahieren sie Daten aus Ihrem Data Warehouse.

Gehen Sie wie folgt vor, um Extraktionsagenten für Ihr AWS SCT Projekt zu registrieren.

So registrieren Sie einen Agenten für die Datenextraktion

1. Starten Sie AWS Schema Conversion Tool das und öffnen Sie ein Projekt.
2. Öffnen Sie das Menü Ansicht und wählen Sie dann Datenmigrationsansicht (andere). Die Registerkarte Agents wird angezeigt. Wenn Sie bereits Agenten registriert haben, AWS SCT werden diese in einem Raster oben auf der Registerkarte angezeigt.
3. Wählen Sie Register aus.

Nachdem Sie einen Agenten für ein AWS SCT Projekt registriert haben, können Sie denselben Agenten nicht mehr für ein anderes Projekt registrieren. Wenn Sie in einem AWS SCT Projekt keinen Agenten mehr verwenden, können Sie die Registrierung für ihn aufheben. Sie können ihn dann für ein anderes Projekt registrieren.

4. Wählen Sie Redshift Data Agent und dann OK.
5. Geben Sie Ihre Informationen auf der Registerkarte Verbindung des Dialogfelds ein:
 - a. Geben Sie unter Beschreibung eine Beschreibung des Agenten ein.
 - b. Geben Sie als Hostname den Hostnamen oder die IP-Adresse des Computers des Agenten ein.
 - c. Geben Sie für Port die Portnummer ein, die der Agent abhört.
 - d. Wählen Sie Registrieren, um den Agenten bei Ihrem AWS SCT Projekt zu registrieren.
6. Wiederholen Sie die obigen Schritte, um mehrere Agenten in Ihrem AWS SCT -Projekt zu registrieren.

Informationen für einen AWS SCT Agenten ausblenden und wiederherstellen

Ein AWS SCT Agent verschlüsselt eine beträchtliche Menge an Informationen, z. B. Kennwörter für Schlüsselspeicher von Benutzern, Datenbankkonten, AWS Kontoinformationen und ähnliche Elemente. Dazu wird eine spezielle Datei mit dem Namen `seed.dat` eingesetzt. Standardmäßig erstellt der Agent die Datei im Arbeitsordner des Benutzers, der zuerst den Agenten konfiguriert.

Da unterschiedliche Benutzer den Agenten konfigurieren und ausführen können, wird der Pfad zur Datei `seed.dat` im Parameter `{extractor.private.folder}` der Datei `settings.properties` gespeichert. Wenn der Agent gestartet wird, nutzt er diesen Pfad, um die Datei `seed.dat` zu finden, damit er auf die Schlüssel-Vertrauensspeicherdaten für die Datenbank zugreifen kann, mit der er arbeitet.

In den folgenden Fällen müssen Sie vom Agenten gespeicherte Passwörter möglicherweise wiederherstellen:

- Wenn der Benutzer die `seed.dat` Datei verliert und der Standort und der Port des AWS SCT Agenten sich nicht geändert haben.
- Wenn der Benutzer die `seed.dat` Datei verliert und sich der Standort und der Port des AWS SCT Agenten geändert haben. In diesem Fall tritt die Änderung in der Regel auf, da der Agent zu einem anderen Host oder Port migriert wurde und die Informationen in der Datei `seed.dat` nicht mehr gültig sind.

Wenn in diesen Fällen der Agent ohne SSL gestartet wurde, startet der Agent und greift auf den zuvor erstellten Agentenspeicher zu. Anschließend wechselt er in den Status `Waiting for recovery` (Auf Wiederherstellung warten).

Wenn der Agent in diesen Fällen jedoch mit SSL gestartet wurde, können Sie ihn nicht erneut starten. Der Grund hierfür ist, dass der Agent die Passwörter für die in der Datei `settings.properties` gespeicherten Zertifikate nicht entschlüsseln kann. Bei dieser Art des Startens schlägt der Agentenstart fehl. Im Protokoll wird eine Fehlermeldung ähnlich der folgenden angezeigt: "Der Agent kann mit aktivierten SSL-Modus nicht gestartet werden. Konfigurieren Sie den Agenten neu. Grund: Das Passwort für den Schlüsselspeicher ist nicht korrekt."

Um dieses Problem zu beheben, erstellen Sie einen neuen Agenten und konfigurieren diesen so, dass er für den Zugriff auf SSL-Zertifikate die vorhandenen Passwörter nutzt. Führen Sie dazu die folgenden Schritte aus.

Nachdem Sie dieses Verfahren ausgeführt haben, sollte der Agent ausgeführt werden und in den Status Warten auf Wiederherstellung wechseln. AWS SCT sendet automatisch die benötigten Passwörter an einen Agenten mit dem Status Warten auf Wiederherstellung. Wenn der Agent über die neuen Passwörter verfügt, startet er alle Aufgaben neu. AWS SCT Nebenbei ist keine weitere Benutzeraktion erforderlich.

So konfigurieren Sie den Agenten neu und stellen Passwörter für den Zugriff auf SSL-Zertifikate wieder her

1. Installieren Sie einen neuen AWS SCT Agenten und führen Sie die Konfiguration aus.
2. Ändern Sie die `agent.name`-Eigenschaft in der Datei `instance.properties` auf den Namen des Agenten, für den der Speicher erstellt wurde, damit der neue Agent den vorhandenen Agentenspeicher nutzt.

Die Datei `instance.properties` ist im privaten Ordner des Agenten gespeichert, der nach der folgenden Konvention benannt wird: `{output.folder}\dmt\{hostName}_{portNumber}\`.

3. Ändern Sie den Namen von `{output.folder}` in den Namen des vorherigen Ausgabeordners des Agenten.

Derzeit versucht er immer AWS SCT noch, auf den alten Extractor auf dem alten Host und Port zuzugreifen. Deshalb erhält das Extraktionsprogramm, auf das nicht zugegriffen werden kann, den Status FAILED. Anschließend können Sie Host und Port ändern.

4. Ändern Sie den Host und/oder Port für den alten Agenten über den Befehl zum Modifizieren, um den Anforderungs-Flow zum neuen Agenten umzuleiten.

Wann AWS SCT kann der neue Agent pingen, AWS SCT erhält vom Agenten den Status Wartet auf Wiederherstellung. AWS SCT stellt dann automatisch die Passwörter für den Agenten wieder her.

Jeder Agent, der den Agentenspeicher nutzt, aktualisiert eine spezielle Datei namens `storage.lock`, die im `{output.folder}\{agentName}\storage\` gespeichert ist. Diese Datei enthält die Netzwerk-ID des Agenten und die Uhrzeit, bis zu der der Speicher gesperrt ist. Wenn der Agent den Agentenspeicher nutzt, aktualisiert er die Datei `storage.lock` und erweitert das Speicher-Lease alle 5 Minuten um 10 Minuten. Vor Lease-Ablauf kann keine andere Instance diesen Agentenspeicher nutzen.

Regeln für die Datenmigration erstellen in AWS SCT

Bevor Sie Ihre Daten mit dem extrahieren AWS Schema Conversion Tool, können Sie Filter einrichten, die die Menge der extrahierten Daten reduzieren. Sie können Datenmigrationsregeln erstellen, indem Sie WHERE Klauseln verwenden, um die Menge der zu extrahierenden Daten zu reduzieren. Sie können beispielsweise eine WHERE-Klausel schreiben, die Daten aus einer einzelnen Tabelle auswählt.

Sie können Datenmigrationsregeln erstellen und die Filter als Teil Ihres Projekts speichern. Gehen Sie bei geöffnetem Projekt wie folgt vor, um Datenmigrationsregeln zu erstellen.

Um Regeln für die Datenmigration zu erstellen

1. Öffnen Sie das Menü Ansicht und wählen Sie dann Datenmigrationsansicht (andere).
2. Wählen Sie Datenmigrationsregeln und dann Neue Regel hinzufügen aus.
3. Konfigurieren Sie Ihre Datenmigrationsregel:
 - a. Geben Sie unter Name einen Namen für Ihre Datenmigrationsregel ein.
 - b. Geben Sie unter Where schema name is like einen Filter ein, der auf Schemas angewendet werden soll. In diesem Filter wird eine WHERE-Klausel mithilfe einer LIKE-Klausel ausgewertet. Um ein Schema auszuwählen, geben Sie einen exakten Schemanamen ein. Um mehrere Schemas auszuwählen, verwenden Sie das Zeichen „%“ als Platzhalter für eine beliebige Anzahl von Zeichen im Schemanamen.
 - c. Geben Sie für Tabellennamen wie einen Filter ein, der auf Tabellen angewendet werden soll. In diesem Filter wird eine WHERE-Klausel mithilfe einer LIKE-Klausel ausgewertet. Um eine Tabelle auszuwählen, geben Sie einen exakten Namen ein. Um mehrere Tabellen auszuwählen, verwenden Sie das Zeichen „%“ als Platzhalter für eine beliebige Anzahl von Zeichen im Tabellennamen.
 - d. Geben Sie für Where-Klausel eine WHERE Klausel ein, um Daten zu filtern.
4. Nachdem Sie Ihren Filter konfiguriert haben, wählen Sie Save, um Ihren Filter zu speichern, oder Cancel, um die Änderungen zu verwerfen.
5. Wenn Sie mit dem Hinzufügen, Bearbeiten und Löschen von Filtern fertig sind, wählen Sie Alle speichern, um alle Ihre Änderungen zu speichern.

Mit dem Umschaltsymbol lässt sich ein Filter deaktivieren, ohne ihn zu löschen. Das Kopiersymbol verwenden Sie, um einen vorhandenen Filter zu duplizieren. Mit dem Löschsymboll können Sie einen

vorhandenen Filter löschen. Um alle Änderungen zu speichern, die Sie an Ihren Filtern vornehmen, wählen Sie Alle speichern.

Ändern der Extraktor- und Kopiereinstellungen aus den Projekteinstellungen

Im Fenster mit den Projekteinstellungen in AWS SCT können Sie Einstellungen für Datenextraktionsagenten und den Amazon Redshift COPY Redshift-Befehl auswählen.

Um diese Einstellungen auszuwählen, wählen Sie Einstellungen, Projekteinstellungen und dann Datenmigration. Hier können Sie die Extraktionseinstellungen, Amazon S3 S3-Einstellungen und Kopiereinstellungen bearbeiten.

Verwenden Sie die Anweisungen in der folgenden Tabelle, um die Informationen für die Extraktionseinstellungen bereitzustellen.

Für diesen Parameter	Vorgehensweise
Compression format (Komprimierungsformat)	Geben Sie das Komprimierungsformat der Eingabedateien an. Wählen Sie eine der folgenden Optionen: GZIP, BZIP2, ZSTD oder Keine Komprimierung.
Trennzeichen	Geben Sie das ASCII-Zeichen an, das Felder in den Eingabedateien trennt. Nicht druckbare Zeichen werden nicht unterstützt.
NULL-Wert als Zeichenfolge	Aktivieren Sie diese Option, wenn Ihre Daten ein Nullabschlusssymbol enthalten. Wenn diese Option ausgeschaltet ist, behandelt der Amazon Redshift COPY Redshift-Befehl Null als Ende des Datensatzes und beendet den Ladevorgang.
Strategie zur Sortierung	Verwenden Sie die Sortierung, um die Extraktion an der Fehlerstelle neu zu starten. Wählen Sie eine der folgenden Sortierstrategien: Verwenden Sie die Sortierung nach dem ersten Fehler (empfohlen), Verwenden Sie die Sortierung, wenn möglich, oder Verwenden Sie die Sortierung nie. Weitere Informationen finden Sie unter the section called "Daten sortieren" .

Für diesen Parameter	Vorgehensweise
Temporäres Quellschema	Geben Sie den Namen des Schemas in die Quelldatenbank ein, in der der Extraktionsagent die temporären Objekte erstellen kann.
Größe der Ausgangsdatei (in MB)	Geben Sie die Größe der auf Amazon S3 hochgeladenen Dateien in MB ein.
Größe der Snowball-Out-Datei (in MB)	Geben Sie die Größe der hochgeladenen Dateien in MB ein. AWS Snowball Dateien können 1—1.000 MB groß sein.
Verwenden Sie die automatische Partitionierung. Geben Sie für Greenplum und Netezza die Mindestgröße der unterstützten Tabellen ein (in Megabyte)	Aktivieren Sie diese Option, um die Tabellenpartitionierung zu verwenden, und geben Sie dann die Größe der zu partitionierenden Tabellen für Greenplum- und Netezza-Quelldatenbanken ein. Bei Migrationen von Oracle zu Amazon Redshift können Sie dieses Feld leer lassen, da Unteraufgaben für alle AWS SCT partitionierten Tabellen erstellt werden.
LOBs extrahieren	Aktivieren Sie diese Option, um große Objekte (LOBs) aus Ihrer Quelldatenbank zu extrahieren. Zu den LOBs gehören BLOBs, CLOBs, NCLOBs, XML-Dateien usw. AWS SCT Extraktionsagenten erstellen für jedes LOB eine Datendatei.
Amazon S3 S3-Bucket-LOBs-Ordner	Geben Sie den Speicherort für AWS SCT Extraktionsagenten zum Speichern von LOBs ein.
Wenden Sie RTRIM auf Zeichenkettenspalten an	Aktivieren Sie diese Option, um einen bestimmten Zeichensatz vom Ende der extrahierten Zeichenketten abzuschneiden.
Bewahren Sie Dateien nach dem Upload auf Amazon S3 lokal auf	Aktivieren Sie diese Option, um Dateien auf Ihrem lokalen Computer zu behalten, nachdem Datenextraktionsagenten sie auf Amazon S3 hochgeladen haben.

Verwenden Sie die Anweisungen in der folgenden Tabelle, um die Informationen für die Amazon S3 S3-Einstellungen bereitzustellen.

Für diesen Parameter	Vorgehensweise
Verwenden Sie einen Proxy	Aktivieren Sie diese Option, um einen Proxy-Server zum Hochladen von Daten auf Amazon S3 zu verwenden. Wählen Sie dann das Datenübertragungsprotokoll und geben Sie den Hostnamen, den Port, den Benutzernamen und das Passwort ein.
Endpunkttyp	Wählen Sie FIPS, um den FIPS-Endpunkt (Federal Information Processing Standard) zu verwenden. Wählen Sie VPCE, um den Virtual Private Cloud (VPC) -Endpunkt zu verwenden. Geben Sie dann für VPC-Endpunkt das Domain Name System (DNS) Ihres VPC-Endpunkts ein.
Dateien nach dem Kopieren nach Amazon Redshift auf Amazon S3 behalten	Aktivieren Sie diese Option, um die extrahierten Dateien auf Amazon S3 zu behalten, nachdem Sie diese Dateien nach Amazon Redshift kopiert haben.

Verwenden Sie die Anweisungen in der folgenden Tabelle, um die Informationen für die Kopiereinstellungen bereitzustellen.

Für diesen Parameter	Vorgehensweise
Maximale Anzahl von Fehlern	Geben Sie die Anzahl der Ladefehler ein. Wenn der Vorgang dieses Limit erreicht hat, beenden die AWS SCT Datenextraktionsagenten den Datenladevorgang. Der Standardwert ist 0, was bedeutet, dass die AWS SCT Datenextraktionsagenten den Datenladevorgang unabhängig von den Fehlern fortsetzen.
Ersetzt ungültige UTF-8-Zeichen	Aktivieren Sie diese Option, um ungültige UTF-8-Zeichen durch das angegebene Zeichen zu ersetzen und den Datenladevorgang fortzusetzen.
Verwenden Sie das Leerzeichen als Nullwert	Aktivieren Sie diese Option, um leere Felder, die aus Leerzeichen bestehen, als Null zu laden.

Für diesen Parameter	Vorgehensweise
Verwenden Sie leer als Nullwert	Aktivieren Sie diese Option, um leere VARCHAR Felder CHAR und Felder mit dem Wert Null zu laden.
Spalten kürzen	Aktivieren Sie diese Option, um Daten in Spalten so zu kürzen, dass sie der Datentypspezifikation entsprechen.
Automatische Komprimierung	Aktivieren Sie diese Option, um während eines Kopiervorgangs eine Komprimierungskodierung anzuwenden.
Automatische Aktualisierung der Statistiken	Aktivieren Sie diese Option, um die Statistiken am Ende eines Kopiervorgangs zu aktualisieren.
Überprüfen Sie die Datei vor dem Laden	Aktivieren Sie diese Option, um Datendateien zu validieren, bevor sie in Amazon Redshift geladen werden.

Sortieren von Daten vor der Migration mit AWS SCT

Das Sortieren Ihrer Daten vor der Migration mit AWS SCT bietet einige Vorteile. Wenn Sie die Daten zuerst sortieren, AWS SCT können Sie den Extraktionsagenten nach einem Fehler am zuletzt gespeicherten Punkt neu starten. Wenn Sie Daten zu Amazon Redshift migrieren und die Daten zuerst sortieren, AWS SCT können Sie Daten außerdem schneller in Amazon Redshift einfügen.

Diese Vorteile haben damit zu tun, wie Datenextraktionsabfragen AWS SCT erstellt werden. In einigen Fällen AWS SCT verwendet die Analysefunktion DENSE_RANK in diesen Abfragen. DENSE_RANK kann jedoch viel Zeit und Serverressourcen in Anspruch nehmen, um den Datensatz zu sortieren, der sich aus der Extraktion ergibt. Wenn also auch ohne ihn funktioniert, funktioniert er AWS SCT auch.

Um Daten vor der Migration zu sortieren, verwenden Sie AWS SCT

1. Öffnen Sie ein AWS SCT Projekt.
2. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt und wählen Sie dann Lokale Aufgabe erstellen.
3. Wählen Sie die Registerkarte Erweitert und wählen Sie unter Sortierstrategie eine Option aus:

- Verwenden Sie niemals die Sortierung — Der Extraktionsagent verwendet die Analysefunktion DENSE_RANK nicht und startet bei einem Fehler von vorne neu.
- Verwenden Sie nach Möglichkeit die Sortierung — Der Extraktionsagent verwendet DENSE_RANK, wenn die Tabelle einen Primärschlüssel oder eine eindeutige Einschränkung hat.
- Sortierung nach dem ersten Fehler verwenden (empfohlen) — Der Extraktionsagent versucht zunächst, die Daten abzurufen, ohne DENSE_RANK zu verwenden. Schlägt der erste Versuch fehl, baut der Agent für die Extraktion die Abfrage unter Verwendung von DENSE_RANK um und behält im Fehlerfall ihren Speicherort bei.

Create Local task

General | **Advanced** | Source server | AWS S3 settings | Source SSL settings

Extraction settings

Delimiter character: |

Compression format: GZIP

NULL value as a string

Sorting strategy: Use sorting after first fail (recommen...)

Source temp schema:

Out file size (in MB): 10

Apply RTRIM to string columns

Keep files locally after upload to AWS S3

Use subtasks auto-balancing between agents

Freezing interval: 10

Copy settings

Maximum error count: 0

Replace invalid UTF-8 character: ?

Use blank as null value
BLANKSASNULL: This option loads blank fields, which consist of only white space characters, as NULL. The default behavior, without this option, is to load the space characters as is.

Use empty as null value
EMPTYASNULL: This option indicates that Amazon Redshift should load empty CHAR and VARCHAR fields as NULL.

Truncate columns
TRUNCATECOLUMNS: This option truncates data in columns to the appropriate number of characters so that it fits the column specification. This option applies only to columns with a VARCHAR or CHAR data type, and rows 4 MB or less in size.

Automatic compression
COMPUPDATE: This option controls whether compression encodings are automatically

Test Task | Cancel | Create

- Legen Sie zusätzliche Parameter fest wie folgt, und wählen Sie Create, um Ihre Aufgabe zum Extrahieren Ihrer Daten zu erstellen.

Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen

Gehen Sie wie folgt vor, um Datenextraktionsaufgaben zu erstellen, auszuführen und zu überwachen.

So weisen Sie Agenten Aufgaben zu und migrieren Daten

1. Nachdem Sie Ihr Schema konvertiert haben, wählen Sie im linken Bereich Ihres Projekts eine oder mehrere Tabellen aus. AWS Schema Conversion Tool

Sie können alle Tabellen auswählen. Um die Leistung nicht zu sehr zu beeinträchtigen, raten wir jedoch von dieser Vorgehensweise ab. Wir empfehlen, für mehrere Tabellen mehrere Aufgaben basierend auf der Größe der Tabellen in Ihrem Data Warehouse zu erstellen.

2. Öffnen Sie das Kontextmenü (Rechtsklick) für jede Tabelle und wählen Sie dann Aufgabe erstellen. Das Dialogfeld Lokale Aufgabe erstellen wird geöffnet.
3. Geben Sie unter Aufgabenname einen Namen für die Aufgabe ein.
4. Wählen Sie für den Migrationsmodus eine der folgenden Optionen:
 - Nur Extrahieren — Extrahieren Sie Ihre Daten und speichern Sie die Daten in Ihren lokalen Arbeitsordnern.
 - Extrahieren und hochladen — Extrahieren Sie Ihre Daten und laden Sie Ihre Daten auf Amazon S3 hoch.
 - Extrahieren, Hochladen und Kopieren — Extrahieren Sie Ihre Daten, laden Sie Ihre Daten auf Amazon S3 hoch und kopieren Sie sie in Ihr Amazon Redshift Data Warehouse.
5. Wählen Sie als Verschlüsselungstyp eine der folgenden Optionen aus:
 - KEINE — Deaktiviert die Datenverschlüsselung für den gesamten Datenmigrationsprozess.
 - CSE_SK — Verwenden Sie eine clientseitige Verschlüsselung mit einem symmetrischen Schlüssel, um Daten zu migrieren. AWS SCT generiert automatisch Verschlüsselungsschlüssel und überträgt sie mithilfe von Secure Sockets Layer (SSL) an Datenextraktionsagenten. AWS SCT verschlüsselt keine großen Objekte (LOBs) während der Datenmigration.
6. Wählen Sie Extract LOBs, um große Objekte zu extrahieren. Wenn Sie keine großen Objekte extrahieren müssen, können Sie das Kontrollkästchen deaktivieren. Dadurch verringert sich die Menge der Daten, die Sie extrahieren.
7. Um detaillierte Informationen zu einer Aufgabe anzuzeigen, wählen Sie Aufgabenprotokollierung aktivieren. Sie können das Aufgabenprotokoll zum Debuggen von Problemen verwenden.

Wenn Sie die Aufgabenprotokollierung aktivieren, wählen Sie die gewünschte Detailebene aus. Es sind folgende Ebenen verfügbar, wobei jede Ebene alle Nachrichten der darunterliegenden Ebene(n) enthält:

- ERROR— Die kleinste Menge an Details.
 - WARNING
 - INFO
 - DEBUG
 - TRACE— Die größte Menge an Details.
8. AWS SCT Verwendet zum Exportieren von BigQuery Daten den Bucket-Ordner von Google Cloud Storage. In diesem Ordner speichern Datenextraktionsagenten Ihre Quelldaten.

Um den Pfad zu Ihrem Google Cloud Storage-Bucket-Ordner einzugeben, wählen Sie Erweitert. Geben Sie für den Google CS-Bucket-Ordner den Bucket-Namen und den Ordnernamen ein.

9. Um eine Rolle für den Benutzer Ihres Datenextraktionsagenten zu übernehmen, wählen Sie Amazon S3 S3-Einstellungen. Geben Sie für die IAM-Rolle den Namen der zu verwendenden Rolle ein. Wählen Sie für Region die AWS-Region für diese Rolle aus.
10. Wählen Sie Testaufgabe, um zu überprüfen, ob Sie eine Verbindung zu Ihrem Arbeitsordner, Amazon S3 S3-Bucket und Amazon Redshift Data Warehouse herstellen können. Die Überprüfung ist abhängig vom gewählten Migrationsmodus.
11. Klicken Sie auf Create, um die Aufgabe zu erstellen.
12. Wiederholen Sie die vorherigen Schritte, um Aufgaben für alle zu migrierenden Daten zu erstellen.

So können Sie Aufgaben ausführen und überwachen

1. Wählen Sie für Ansicht die Datenmigrationsansicht aus. Die Registerkarte Agents wird angezeigt.
2. Wählen Sie die Registerkarte Tasks aus. Ihre Aufgaben werden im Raster oben wie folgt angezeigt. Der Status einer Aufgabe wird im oberen Raster und der Status der Unteraufgaben im unteren Raster angezeigt.

The screenshot displays the AWS Schema Conversion Tool interface. At the top, there are tabs for Agents, Tasks, Virtual partitions, Data migration rules, and Snowball. The main area shows a table of tasks with columns for Name, Extract, Upload, and Copy. The tasks listed are CUSTOMER, LINEORDER_100K, LINEORDER_150K, LINEORDER_1M, LocalTask 2, CUSTOMER, LINEORDER_100K, LINEORDER_150K, LocalTask 3, and LINEORDER_100K. The progress bars for Extract, Upload, and Copy are shown for each task. LocalTask 2 and LocalTask 3 are expanded, showing their sub-tasks. The LocalTask 3 sub-task LINEORDER_100K has 0% progress in the Copy column. Below the table, there is a toolbar with buttons for Download log, All migration Tasks, Resume, Stop, Restart, Reset, Delete, Replace, Refresh all, and Refresh selected. At the bottom, there are tabs for Properties and Processing details, and a progress bar for the selected task showing 100% completion.

3. Wählen Sie oben im Raster eine Aufgabe aus und erweitern Sie diese. Abhängig vom gewählten Migrationsmodus ist die Aufgabe in die Punkte Extract, Upload und Copy unterteilt.
4. Wählen Sie Start für eine Aufgabe, um sie zu starten. Sie können den Status Ihrer Aufgaben während der Ausführung überwachen. Die Unteraufgaben werden parallel ausgeführt. Auch das Extrahieren, Hochladen und Kopieren erfolgt parallel.
5. Wenn Sie beim Einrichten der Aufgabe die Protokollierung aktiviert haben, können Sie das Protokoll anzeigen:
 - a. Wählen Sie Protokoll herunterladen. Es wird eine Nachricht mit dem Namen des Verzeichnisses angezeigt, das die Protokolldatei enthält. Schließen Sie die Nachricht.
 - b. Auf der Registerkarte Task details wird ein Link angezeigt. Klicken Sie auf den Link, um das Verzeichnis zu öffnen, das die Protokolldatei enthält.

Sie können schließen AWS SCT und Ihre Agenten und Aufgaben werden weiterhin ausgeführt. Sie können das Programm AWS SCT später erneut öffnen, um den Status Ihrer Aufgaben zu überprüfen und die Task-Protokolle einzusehen.

Sie können Datenextraktionsaufgaben auf Ihrer lokalen Festplatte speichern und sie mithilfe von Export und Import in demselben oder einem anderen Projekt wiederherstellen. Um eine Aufgabe zu exportieren, stellen Sie sicher, dass in einem Projekt mindestens eine Extraktionsaufgabe erstellt

wurde. Sie können eine einzelne Extraktionsaufgabe oder alle im Projekt erstellten Aufgaben importieren.

Wenn Sie eine Extraktionsaufgabe exportieren, AWS SCT erstellt eine separate .xml Datei für diese Aufgabe. In der .xml Datei werden die Metadateninformationen dieser Aufgabe gespeichert, z. B. Aufgabeneigenschaften, Beschreibung und Unteraufgaben. Die .xml Datei enthält keine Informationen zur Verarbeitung einer Extraktionsaufgabe. Informationen wie die folgenden werden beim Import der Aufgabe neu erstellt:

- Fortschritt der Aufgabe
- Status von Unteraufgaben und Phasen
- Verteilung der Extraktionsmittel nach Unteraufgaben und Stufen
- Aufgaben- und Unteraufgaben-IDs
- Task name

Exportieren und Importieren einer AWS SCT Datenextraktionsaufgabe

Mithilfe von AWS SCT Export und Import können Sie schnell eine vorhandene Aufgabe aus einem Projekt speichern und sie in einem anderen Projekt (oder demselben Projekt) wiederherstellen. Gehen Sie wie folgt vor, um Datenextraktionsaufgaben zu exportieren und zu importieren.

Um eine Datenextraktionsaufgabe zu exportieren und zu importieren

1. Wählen Sie für Ansicht die Datenmigrationsansicht aus. Die Registerkarte Agents wird angezeigt.
2. Wählen Sie die Registerkarte Tasks aus. Ihre Aufgaben werden in dem sich öffnenden Raster aufgeführt.
3. Wählen Sie die drei vertikal ausgerichteten Punkte (Ellipsensymbol) in der unteren rechten Ecke unter der Aufgabenliste aus.
4. Wählen Sie im Popup-Menü die Option „Aufgabe exportieren“.
5. Wählen Sie den Ordner aus, in dem Sie die .xml Aufgabenexportdatei ablegen möchten AWS SCT .

AWS SCT erstellt die Aufgabenexportdatei mit dem Dateinamenformat **TASK-DESCRIPTION_TASK-ID.xml**.

- Wählen Sie die drei vertikal ausgerichteten Punkte (Ellipsensymbol) unten rechts unter der Aufgabenliste aus.
- Wählen Sie im Popup-Menü die Option „Aufgabe importieren“.

Sie können eine Extraktionsaufgabe in ein Projekt importieren, das mit der Quelldatenbank verbunden ist, und das Projekt hat mindestens einen aktiven registrierten Extraktionsagenten.

- Wählen Sie die .xml Datei für die Extraktionsaufgabe aus, die Sie exportiert haben.

AWS SCT ruft die Parameter der Extraktionsaufgabe aus der Datei ab, erstellt die Aufgabe und fügt sie den Extraktionsagenten hinzu.

- Wiederholen Sie diese Schritte, um zusätzliche Datenextraktionsaufgaben zu exportieren und zu importieren.

Am Ende dieses Vorgangs sind Ihr Export und Import abgeschlossen und Ihre Datenextraktionsaufgaben sind einsatzbereit.

Datenextraktion mit einem AWS Snowball Edge-Gerät

Die Verwendung AWS SCT von AWS Snowball Edge besteht aus mehreren Schritten. Die Migration umfasst eine lokale Aufgabe, bei der ein Datenextraktionsagent AWS SCT verwendet wird, um die Daten auf das AWS Snowball Edge-Gerät zu verschieben, und anschließend eine Zwischenaktion, bei der die Daten vom AWS Snowball Edge-Gerät in einen Amazon S3 S3-Bucket AWS kopiert werden. Der Vorgang schließt das AWS SCT Laden der Daten aus dem Amazon S3 S3-Bucket nach Amazon Redshift ab.

Die Abschnitte, die dieser Übersicht folgen, enthalten eine step-by-step Anleitung zu jeder dieser Aufgaben. Das Verfahren setzt voraus, dass Sie einen Datenextraktionsagenten auf einem speziellen Computer AWS SCT installiert und konfiguriert und registriert haben.

Führen Sie die folgenden Schritte aus, um Daten mithilfe von AWS Snowball Edge von einem lokalen Datenspeicher zu einem AWS Datenspeicher zu migrieren.

- Erstellen Sie mit der AWS Snowball Konsole einen AWS Snowball Edge-Job.
- Entsperren Sie das AWS Snowball Edge-Gerät mithilfe des lokalen, dedizierten Linux-Computers.
- Erstellen Sie ein neues Projekt in AWS SCT.
- Installieren und konfigurieren Sie Ihre Datenextraktionsagenten.

5. Erstellen Sie Berechtigungen für den zu verwendenden Amazon S3-Bucket und richten Sie diese ein.
6. Importieren Sie einen AWS Snowball Job in Ihr AWS SCT Projekt.
7. Registrieren Sie Ihren Datenextraktionsagenten in AWS SCT.
8. Erstellen Sie eine lokale Aufgabe in AWS SCT.
9. Führen Sie die Datenmigrationsaufgabe in aus und überwachen Sie sie AWS SCT.

tep-by-step S-Verfahren für die Datenmigration mithilfe von AWS SCT Edge AWS Snowball

Die folgenden Abschnitte enthalten ausführliche Informationen über die einzelnen Migrationsschritte.

Schritt 1: Erstellen Sie einen AWS Snowball Edge-Job

Erstellen Sie einen AWS Snowball Job, indem Sie die im Abschnitt [Erstellen eines AWS Snowball Edge-Jobs im Edge-Entwicklerhandbuch](#) beschriebenen Schritte ausführen. AWS Snowball

Schritt 2: Entsperren Sie das AWS Snowball Edge-Gerät

Führen Sie die Befehle zum Entsperren und Bereitstellen von Anmeldeinformationen für das Snowball Edge-Gerät von dem Computer aus, auf dem Sie den AWS DMS Agenten installiert haben. Wenn Sie diese Befehle ausführen, können Sie sicherstellen, dass der AWS DMS Agentenanruf eine Verbindung zum AWS Snowball Edge-Gerät herstellt. Weitere Informationen zum Entsperren des AWS Snowball Edge-Geräts finden Sie unter [Entsperren des Snowball Edge](#).

```
aws s3 ls s3://<bucket-name> --profile <Snowball Edge profile> --endpoint http://  
<Snowball IP>:8080 --recursive
```

Schritt 3: Erstellen Sie ein neues Projekt AWS SCT

Als Nächstes erstellen Sie ein neues AWS SCT Projekt.

Um ein neues Projekt zu erstellen in AWS SCT

1. Starte das AWS Schema Conversion Tool. Wählen Sie im Menü Datei die Option Neues Projekt aus. Das Dialogfeld „Neues Projekt“ wird angezeigt.
2. Geben Sie einen Namen für Ihr Projekt ein, das lokal auf Ihrem Computer gespeichert wird.

3. Geben Sie den Speicherort für Ihre lokale Projektdatei an.
4. Wählen Sie OK, um Ihr AWS SCT Projekt zu erstellen.
5. Wählen Sie Quelle hinzufügen, um Ihrem AWS SCT Projekt eine neue Quelldatenbank hinzuzufügen.
6. Wählen Sie Ziel hinzufügen, um Ihrem AWS SCT Projekt eine neue Zielplattform hinzuzufügen.
7. Wählen Sie im linken Bereich das Quelldatenbankschema aus.
8. Geben Sie im rechten Bereich die Zieldatenbankplattform für das ausgewählte Quellschema an.
9. Wählen Sie Zuordnung erstellen. Diese Schaltfläche wird aktiv, nachdem Sie das Quelldatenbankschema und die Zieldatenbankplattform ausgewählt haben.

Schritt 4: Installieren und konfigurieren Sie Ihren Datenextraktionsagenten

AWS SCT verwendet einen Datenextraktionsagenten, um Daten zu Amazon Redshift zu migrieren. Die ZIP-Datei, die Sie zur Installation heruntergeladen haben AWS SCT, enthält die Installationsdatei für den Extraktionsagenten. Sie können den Datenextraktionsagenten unter Windows, Red Hat Enterprise Linux oder Ubuntu installieren. Weitere Informationen finden Sie unter [Installation von Extraktionsagenten](#).

Um Ihren Datenextraktionsagenten zu konfigurieren, geben Sie Ihre Quell- und Zieldatenbank-Engines ein. Stellen Sie außerdem sicher, dass Sie die JDBC-Treiber für Ihre Quell- und Zieldatenbanken auf den Computer heruntergeladen haben, auf dem Sie Ihren Datenextraktionsagenten ausführen. Datenextraktionsagenten verwenden diese Treiber, um eine Verbindung zu Ihren Quell- und Zieldatenbanken herzustellen. Weitere Informationen finden Sie unter [Die erforderlichen Datenbanktreiber werden heruntergeladen](#).

In Windows startet das Installationsprogramm für den Datenextraktionsagenten den Konfigurationsassistenten im Befehlszeilenfenster. Unter Linux führen Sie die `sct-extractor-setup.sh` Datei von dem Ort aus, an dem Sie den Agenten installiert haben.

Schritt 5: Konfiguration AWS SCT für den Zugriff auf den Amazon S3 S3-Bucket

Informationen zur Konfiguration eines Amazon S3 [S3-Buckets finden Sie unter Buckets-Übersicht](#) im Amazon Simple Storage Service-Benutzerhandbuch.

Schritt 6: Importieren Sie einen AWS Snowball Job in Ihr Projekt AWS SCT

Importieren Sie Ihren AWS Snowball Job, um Ihr AWS SCT Projekt mit Ihrem AWS Snowball Edge-Gerät zu verbinden.

Um deinen AWS Snowball Job zu importieren

1. Öffnen Sie das Einstellungsmenü und wählen Sie dann Allgemeine Einstellungen. Das Dialogfeld Global settings wird angezeigt.
2. Wählen Sie AWS Dienstprofile und dann Job importieren aus.
3. Wählen Sie Ihren AWS Snowball Job aus.
4. Geben Sie Ihre AWS Snowball IP ein. Weitere Informationen finden Sie im AWS Snowball Benutzerhandbuch unter [Ändern Ihrer IP-Adresse](#).
5. Geben Sie Ihren AWS Snowball Port ein. Weitere Informationen finden Sie unter Für die [Nutzung von AWS Diensten auf einem AWS Snowball Edge-Gerät erforderliche Ports](#) im AWS Snowball Edge-Entwicklerhandbuch.
6. Geben Sie Ihren AWS Snowball Zugriffsschlüssel und Ihren AWS Snowball geheimen Schlüssel ein. Weitere Informationen finden Sie unter [Autorisierung und Zugriffskontrolle AWS Snowball im AWS Snowball](#) Benutzerhandbuch.
7. Wählen Sie Übernehmen und anschließend OK aus.

Schritt 7: Registrieren Sie einen Datenextraktionsagenten in AWS SCT

In diesem Abschnitt registrieren Sie den Datenextraktionsagenten in AWS SCT.

Um einen Datenextraktionsagenten zu registrieren

1. Wählen Sie im Menü Ansicht die Option Datenmigrationsansicht (andere) und dann Registrieren aus.
2. Geben Sie unter Beschreibung einen Namen für Ihren Datenextraktionsagenten ein.
3. Geben Sie als Hostname die IP-Adresse des Computers ein, auf dem Sie Ihren Datenextraktionsagenten ausführen.
4. Geben Sie unter Port den Listening-Port ein, den Sie konfiguriert haben.
5. Wählen Sie Register aus.

Schritt 8: Erstellen einer lokalen Aufgabe

Als Nächstes erstellen Sie die Migrationsaufgabe. Die Aufgabe umfasst zwei Unteraufgaben. Eine Unteraufgabe migriert Daten aus der Quelldatenbank zur AWS Snowball Edge-Appliance. Die andere

Unteraufgabe verschiebt die von der Appliance geladenen Daten in einen Amazon S3-Bucket und migriert sie in die Zieldatenbank.

Um die Migrationsaufgabe zu erstellen

1. Wählen Sie im Menü Ansicht die Option Datenmigrationsansicht (andere) aus.
2. Wählen Sie im linken Bereich, der das Schema Ihrer Quelldatenbank anzeigt, ein Schemaobjekt aus, das migriert werden soll. Öffnen Sie das Kontextmenü (mit der rechten Maustaste) für das Objekt und wählen Sie dann Lokale Aufgabe erstellen.
3. Geben Sie unter Aufgabenname einen aussagekräftigen Namen für Ihre Datenmigrationsaufgabe ein.
4. Wählen Sie für den Migrationsmodus Extrahieren, Hochladen und Kopieren aus.
5. Wählen Sie Amazon S3 S3-Einstellungen.
6. Wählen Sie Snowball verwenden aus.
7. Geben Sie Ordner und Unterordner in Ihrem Amazon S3 S3-Bucket ein, in denen der Datenextraktionsagent Daten speichern kann.
8. Klicken Sie auf Create, um die Aufgabe zu erstellen.

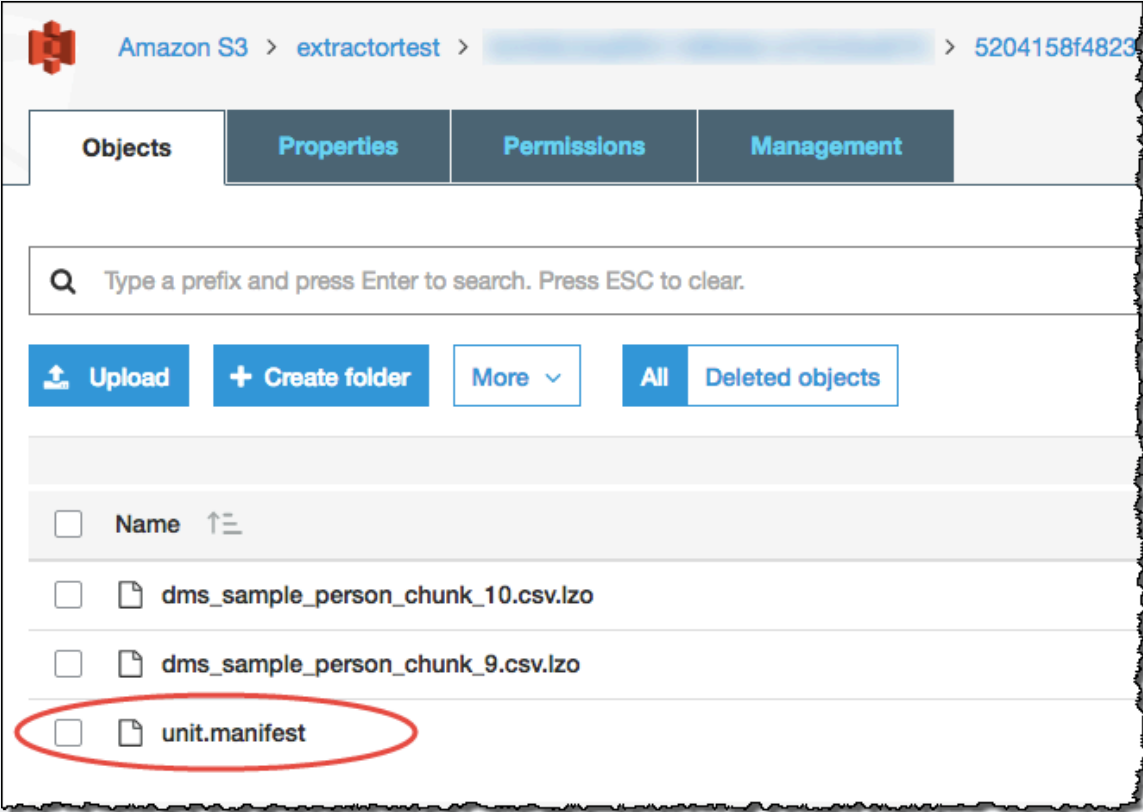
Schritt 9: Ausführen und Überwachen der Datenmigrationsaufgabe in AWS SCT

Um Ihre Datenmigrationsaufgabe zu starten, wählen Sie Start. Stellen Sie sicher, dass Sie Verbindungen zur Quelldatenbank, zum Amazon S3 S3-Bucket, zum AWS Snowball Gerät sowie zur Verbindung mit der Zieldatenbank hergestellt haben AWS.

Sie können die Datenmigrationsaufgaben und ihre Unteraufgaben auf der Registerkarte Aufgaben überwachen und verwalten. Sie können den Fortschritt der Datenmigration verfolgen und Ihre Datenmigrationsaufgaben anhalten oder neu starten.

Ausgabe der Aufgabe zur Datenextraktion

Nach Abschluss der Migration sind Ihre Daten bereit. Lesen Sie die folgenden Informationen, um zu bestimmen, wie Sie abhängig vom gewählten Migrationsmodus und dem Speicherort Ihrer Daten fortfahren sollten.

Migration modus	Speicherort der Daten
Extrahieren, hochladen und kopieren	Die Daten befinden sich bereits in Ihrem Amazon Redshift Data Warehouse. Sie können die Daten dort überprüfen und verwenden. Weitere Informationen finden Sie unter Herstellen einer Verbindung zu Clustern über Client-Tools und Code .
Extrahieren und hochladen	<p>Die Extraktionsagenten haben Ihre Daten als Dateien in Ihrem Amazon S3 S3-Bucket gespeichert. Sie können den Amazon Redshift COPY-Befehl verwenden, um Ihre Daten in Amazon Redshift zu laden. Weitere Informationen finden Sie unter Daten aus Amazon S3 laden in der Amazon Redshift Redshift-Dokumentation.</p> <p>Ihr Amazon S3 S3-Bucket enthält mehrere Ordner, die den von Ihnen eingerichteten Extraktionsaufgaben entsprechen. Wenn Sie Ihre Daten in Amazon Redshift laden, geben Sie den Namen der von jeder Aufgabe erstellten Manifestdatei an. Die Manifestdatei wird im Aufgabenordner in Ihrem Amazon S3 S3-Bucket wie folgt angezeigt.</p>  <p>The screenshot shows the Amazon S3 console interface for a bucket named 'extractortest'. The breadcrumb path is 'Amazon S3 > extractortest > 5204158f4823'. There are four tabs: 'Objects', 'Properties', 'Permissions', and 'Management'. Below the tabs is a search bar with the text 'Type a prefix and press Enter to search. Press ESC to clear.' There are four buttons: 'Upload', 'Create folder', 'More', and 'All Deleted objects'. Below the buttons is a list of objects with checkboxes and names: 'Name', 'dms_sample_person_chunk_10.csv.lzo', 'dms_sample_person_chunk_9.csv.lzo', and 'unit.manifest'. The 'unit.manifest' object is circled in red.</p>

Migration modus	Speicherort der Daten
Nur Extrahieren	Die Agenten für die Datenextraktion haben Ihre Daten als Dateien in Ihrem Arbeitsverzeichnis gespeichert. Kopieren Sie Ihre Daten manuell in Ihren Amazon S3 S3-Bucket und fahren Sie dann mit den Anweisungen für Extrahieren und Hochladen fort.

Verwenden Sie virtuelle Partitionierung mit AWS Schema Conversion Tool

Sie können große nicht partitionierte Tabellen häufig am besten verwalten, indem Sie Unteraufgaben anlegen, die virtuelle Partitionen der Tabellendaten mithilfe von Filterregeln erstellen. In AWS SCT können Sie virtuelle Partitionen für Ihre migrierten Daten erstellen. Es gibt drei Partitionstypen, die jeweils für bestimmte Datentypen eingesetzt werden:

- Der RANGE-Partitionstyp arbeitet mit numerischen Datentypen und Datum und Uhrzeit.
- Der LIST-Partitionstyp arbeitet mit numerischen Datentypen, Zeichen sowie Datum und Uhrzeit.
- Der Partitionstyp DATE AUTO SPLIT funktioniert mit numerischen, Datums- und Uhrzeitdatentypen.

AWS SCT validiert die Werte, die Sie für die Erstellung einer Partition angeben. Wenn Sie beispielsweise versuchen, eine Spalte mit dem Datentyp NUMERIC zu partitionieren, aber Werte eines anderen Datentyps angeben, wird ein AWS SCT Fehler ausgegeben.

Wenn Sie Daten AWS SCT zu Amazon Redshift migrieren, können Sie außerdem die native Partitionierung verwenden, um die Migration großer Tabellen zu verwalten. Weitere Informationen finden Sie unter [Native Partitionierung verwenden](#).

Einschränkungen bei der Erstellung einer virtuellen Partitionierung

Dies sind Einschränkungen für die Erstellung einer virtuellen Partition:

- Sie können die virtuelle Partitionierung nur für partitionierte Tabellen verwenden.
- Sie können die virtuelle Partitionierung nur in der Datenmigration-Ansicht verwenden.
- Sie können die Option UNION ALL VIEW für die virtuelle Partitionierung nicht verwenden.

RANGE-Partitionstyp

Der Partitionstyp RANGE partitioniert Daten basierend auf eine Reihe von Spaltenwerten für numerische und Datums- und Uhrzeitdatentypen. Dieser Partitionstyp erstellt eine WHERE-Klausel, und geben Sie den Wertebereich für jede Partition an. Verwenden Sie das Feld Werte, um eine Werteliste für die partitionierte Spalte anzugeben. Sie können Werteinformationen über eine CSV-Datei laden.

Der RANGE-Partitionstyp erstellt Standardpartitionen an beiden Enden der Partitionswerte. Diese Standardpartitionen catch alle Daten ab, die kleiner oder größer als die angegebenen Partitionswerte sind.

Sie können beispielsweise mehrere Partitionen auf der Grundlage eines von Ihnen angegebenen Wertebereichs erstellen. Im folgenden Beispiel sind die Partitionierungswerte für LO_TAX so angegeben, dass mehrere Partitionen erstellt werden.

```
Partition1: WHERE LO_TAX <= 10000.9
Partition2: WHERE LO_TAX > 10000.9 AND LO_TAX <= 15005.5
Partition3: WHERE LO_TAX > 15005.5 AND LO_TAX <= 25005.95
```

So erstellen Sie eine virtuelle RANGE-Partition

1. Öffnen AWS SCT.
2. Wählen Sie den Modus Datenmigrationsansicht (anderer Modus).
3. Wählen Sie die Tabelle, in der Sie virtuelle Partitionierung einrichten möchten. Öffnen Sie das Kontextmenü (Rechtsklick) für die Tabelle und wählen Sie Virtuelle Partitionierung hinzufügen.
4. Geben Sie im Dialogfeld Virtuelle Partitionierung hinzufügen die Informationen wie folgt ein.

Option	Aktion
Typ der Partition	Wählen Sie RANGE. Das Dialogfeld ändert sich abhängig vom ausgewählten Typ.
Name der Spalte	Wählen Sie die Spalte aus, die Sie partitionieren möchten.
Typ der Spalte	Wählen Sie den Datentyp für die Werte in der Spalte.

Option	Aktion
Werte	Fügen Sie neue Werte hinzu, indem Sie die einzelnen Werte in das Feld New Value eingeben, und klicken Sie auf das Pluszeichen, um die Werte hinzuzufügen.
Aus Datei laden	(Optional) Geben Sie den Namen einer .csv-Datei ein, die die Partitionswerte enthält.

5. Wählen Sie OK aus.

LIST-Partitionstyp

Der Partitionstyp LIST partitioniert Daten basierend auf Spaltenwerten für numerische, Zeichen- sowie Datums- und Uhrzeitdatentypen. Dieser Partitionstyp erstellt eine WHERE-Klausel, und Sie geben die Werte für jede Partition an. Verwenden Sie das Feld Werte, um eine Werteliste für die partitionierte Spalte anzugeben. Sie können Werteinformationen über eine CSV-Datei laden.

Sie können beispielsweise mehrere Partitionen basierend auf einem von Ihnen vorgegebenen Wert erstellen. Im folgenden Beispiel sind die Partitionierungswerte für LO_ORDERKEY so angegeben, dass mehrere Partitionen erstellt werden.

```
Partition1: WHERE LO_ORDERKEY = 1
Partition2: WHERE LO_ORDERKEY = 2
Partition3: WHERE LO_ORDERKEY = 3
...
PartitionN: WHERE LO_ORDERKEY = USER_VALUE_N
```

Sie können auch eine Standard-Partition für Werte erstellen, die nicht in den angegebenen enthalten sind.

Sie können den Partitionstyp LIST verwenden, um die Quelldaten zu filtern, wenn Sie bestimmte Werte von der Migration ausschließen möchten. Nehmen wir beispielsweise an, Sie möchten Zeilen mit LO_ORDERKEY = 4 weglassen. Nehmen Sie in diesem Fall den Wert nicht 4 in die Liste der Partitionswerte auf und stellen Sie sicher, dass die Option Andere Werte einbeziehen nicht ausgewählt ist.

So erstellen Sie eine virtuelle LIST-Partition

1. Öffnen AWS SCT.

2. Wählen Sie den Modus Datenmigrationsansicht (anderer Modus).
3. Wählen Sie die Tabelle, in der Sie virtuelle Partitionierung einrichten möchten. Öffnen Sie das Kontextmenü (Rechtsklick) für die Tabelle und wählen Sie Virtuelle Partitionierung hinzufügen.
4. Geben Sie im Dialogfeld Virtuelle Partitionierung hinzufügen die Informationen wie folgt ein.

Option	Aktion
Typ der Partition	Klicken Sie auf LIST.... Das Dialogfeld ändert sich abhängig vom ausgewählten Typ.
Name der Spalte	Wählen Sie die Spalte aus, die Sie partitionieren möchten.
Neuer Wert	Geben Sie hier einen Wert ein, um ihn der Menge der Partitionierungswerte hinzuzufügen.
Andere Werte einbeziehen	Wählen Sie diese Option, um eine Standard-Partition zu erstellen, in der alle Werte gespeichert werden, die den Partitionierungskriterien nicht entsprechen.
Aus Datei laden	(Optional) Geben Sie den Namen einer .csv-Datei ein, die die Partitionswerte enthält.

5. Wählen Sie OK aus.

DATE AUTO SPLIT Partitionstyp

Der Partitionstyp DATE AUTO SPLIT ist eine automatisierte Methode zum Generieren von RANGE-Partitionen. Mit DATA AUTO SPLIT teilen Sie AWS SCT dem Partitionierungsattribut mit, wo es beginnen und enden soll und wie groß der Bereich zwischen den Werten ist. AWS SCT Berechnet dann automatisch die Partitionswerte.

DATA AUTO SPLIT automatisiert einen Großteil der Arbeit, die mit der Erstellung von Bereichspartitionen verbunden ist. Der Kompromiss zwischen der Verwendung dieser Technik und der Bereichspartitionierung besteht darin, wie viel Kontrolle Sie über die Partitions Grenzen benötigen. Beim automatischen Aufteilen werden immer Bereiche gleicher Größe (einheitlich) erstellt. Durch die Bereichspartitionierung können Sie die Größe der einzelnen Bereiche je nach Bedarf für Ihre spezielle Datenverteilung variieren. Sie können sie beispielsweise täglich, wöchentlich, zweiwöchentlich, monatlich usw. verwenden.

```

Partition1: WHERE LO_ORDERDATE >= '1954-10-10' AND LO_ORDERDATE < '1954-10-24'
Partition2: WHERE LO_ORDERDATE >= '1954-10-24' AND LO_ORDERDATE < '1954-11-06'
Partition3: WHERE LO_ORDERDATE >= '1954-11-06' AND LO_ORDERDATE < '1954-11-20'
...
PartitionN: WHERE LO_ORDERDATE >= USER_VALUE_N AND LO_ORDERDATE <= '2017-08-13'

```

So erstellen Sie eine virtuelle DATE AUTO SPLIT-Partition

1. Öffnen AWS SCT.
2. Wählen Sie den Modus Datenmigrationsansicht (anderer Modus).
3. Wählen Sie die Tabelle, in der Sie virtuelle Partitionierung einrichten möchten. Öffnen Sie das Kontextmenü (Rechtsklick) für die Tabelle und wählen Sie Virtuelle Partitionierung hinzufügen.
4. Geben Sie im Dialogfeld Virtuelle Partitionierung hinzufügen die folgenden Informationen ein.

Option	Aktion
Typ der Partition	Wählen Sie DATE AUTO SPLIT. Das Dialogfeld ändert sich abhängig vom ausgewählten Typ.
Name der Spalte	Wählen Sie die Spalte aus, die Sie partitionieren möchten.
Startdatum	Geben Sie ein Anfangsdatum ein.
Enddatum	Geben Sie ein Enddatum ein.
Interval	Geben Sie die Intervalleinheit ein und wählen Sie den Wert für diese Einheit.

5. Wählen Sie OK aus.

Native Partitionierung verwenden

Um die Datenmigration zu beschleunigen, können Ihre Datenextraktionsagenten native Partitionen von Tabellen auf Ihrem Data Warehouse-Quellserver verwenden. AWS SCT unterstützt native Partitionierung für Migrationen von Greenplum, Netezza und Oracle zu Amazon Redshift.

Nachdem Sie ein Projekt erstellt haben, können Sie beispielsweise Statistiken zu einem Schema sammeln und die Größe der für die Migration ausgewählten Tabellen analysieren. Bei Tabellen, die

die angegebene Größe überschreiten, wird der systemeigene Partitionierungsmechanismus AWS SCT ausgelöst.

Um die native Partitionierung zu verwenden

1. Öffnen AWS SCT Sie und wählen Sie Neues Projekt für Datei. Das Dialogfeld „Neues Projekt“ wird angezeigt.
2. Erstellen Sie ein neues Projekt, fügen Sie Ihre Quell- und Zielservers hinzu und erstellen Sie Zuordnungsregeln. Weitere Informationen finden Sie unter [Ein AWS SCT Projekt erstellen](#).
3. Wählen Sie Ansicht und anschließend Hauptansicht.
4. Wählen Sie für die Projekteinstellungen die Registerkarte Datenmigration. Wählen Sie Automatische Partitionierung verwenden. Geben Sie für Greenplum- und Netezza-Quelldatenbanken die Mindestgröße der unterstützten Tabellen in Megabyte ein (z. B. 100). AWS SCT erstellt automatisch separate Migrationsunteraufgaben für jede native Partition, die nicht leer ist. AWS SCT Erstellt für Migrationen von Oracle zu Amazon Redshift Unteraufgaben für alle partitionierten Tabellen.
5. Wählen Sie im linken Bereich, in dem das Schema aus Ihrer Quelldatenbank angezeigt wird, ein Schema aus. Öffnen Sie das Kontextmenü (Rechtsklick) für das Objekt und wählen Sie Statistik sammeln. Bei der Datenmigration von Oracle zu Amazon Redshift können Sie diesen Schritt überspringen.
6. Wählen Sie alle zu migrierenden Tabellen aus.
7. Registrieren Sie die erforderliche Anzahl von Agenten. Weitere Informationen finden Sie unter [Registrierung von Extraktionsagenten mit dem AWS Schema Conversion Tool](#).
8. Erstellen Sie eine Datenextraktionsaufgabe für die ausgewählten Tabellen. Weitere Informationen finden Sie unter [Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen](#).

Prüfen Sie, ob große Tabellen in Unteraufgaben aufgeteilt sind und ob jede Unteraufgabe dem Datensatz entspricht, der einen Teil der Tabelle darstellt, der sich in einem Bereich in Ihrem Quell-Data Warehouse befindet.

9. Starten und überwachen Sie den Migrationsprozess, bis die AWS SCT Datenextraktionsagenten die Migration der Daten aus Ihren Quelltabellen abgeschlossen haben.

Migration von LOBs zu Amazon Redshift

Amazon Redshift unterstützt das Speichern großer binärer Objekte (LOBs) nicht. Wenn Sie jedoch ein oder mehrere LOBs zu Amazon Redshift migrieren müssen, AWS SCT können Sie die Migration durchführen. AWS SCT verwendet dazu einen Amazon S3 S3-Bucket zum Speichern der LOBs und schreibt die URL für den Amazon S3 S3-Bucket in die migrierten Daten, die in Amazon Redshift gespeichert sind.

So migrieren Sie LOBs zu Amazon Redshift

1. Öffnen Sie ein Projekt AWS SCT .
2. Stellen Sie eine Verbindung mit der Quell- und Zieldatenbank her. Aktualisieren Sie die Metadaten aus der Zieldatenbank und stellen Sie sicher, dass die konvertierten Tabellen dort vorhanden sind.
3. Wählen Sie unter Aktionen die Option Lokale Aufgabe erstellen aus.
4. Wählen Sie für den Migrationsmodus eine der folgenden Optionen:
 - Extrahieren und Hochladen, um Ihre Daten zu extrahieren, und laden Sie Ihre Daten auf Amazon S3 hoch.
 - Extrahieren, Hochladen und Kopieren, um Ihre Daten zu extrahieren, laden Sie Ihre Daten auf Amazon S3 hoch und kopieren Sie sie in Ihr Amazon Redshift Data Warehouse.
5. Wählen Sie Amazon S3 S3-Einstellungen.
6. Geben Sie für den Amazon S3 S3-Bucket-LOBs-Ordner den Namen des Ordners in einem Amazon S3 S3-Bucket ein, in dem die LOBs gespeichert werden sollen.

Wenn Sie ein AWS Serviceprofil verwenden, ist dieses Feld optional. AWS SCT kann die Standardeinstellungen Ihres Profils verwenden. Um einen anderen Amazon S3 S3-Bucket zu verwenden, geben Sie den Pfad hier ein.

7. Aktivieren Sie die Option Proxy verwenden, um einen Proxy-Server zum Hochladen von Daten auf Amazon S3 zu verwenden. Wählen Sie dann das Datenübertragungsprotokoll und geben Sie den Hostnamen, den Port, den Benutzernamen und das Passwort ein.
8. Wählen Sie als Endpunkttyp FIPS aus, um den FIPS-Endpunkt (Federal Information Processing Standard) zu verwenden. Wählen Sie VPCE, um den Virtual Private Cloud (VPC) -Endpunkt zu verwenden. Geben Sie dann für VPC-Endpunkt das Domain Name System (DNS) Ihres VPC-Endpunkts ein.

9. Aktivieren Sie die Option Dateien nach dem Kopieren nach Amazon Redshift auf Amazon S3 behalten, um die extrahierten Dateien nach dem Kopieren dieser Dateien nach Amazon Redshift auf Amazon S3 beizubehalten.
10. Klicken Sie auf Create, um die Aufgabe zu erstellen.

Bewährte Methoden und Problembehebung für Datenextraktionsagenten

Nachfolgend finden Sie einige bewährte Methoden und Problemlösungen für die Verwendung von Agenten für die Datenextraktion.

Problem	Vorschläge für die Fehlerbehebung
Die Leistung ist langsam.	Wir empfehlen zur Leistungsverbesserung Folgendes: <ul style="list-style-type: none"> • Installieren Sie mehrere Agenten. • Installieren Sie Agenten auf Computern, die sich in der Nähe Ihres Data Warehouses befinden. • Bearbeiten Sie nicht alle Tabellen in einer einzelnen Agentenaufgabe.
Verzögerungen durch Konflikte	Vermeiden Sie es, dass zu viele Agenten gleichzeitig auf Ihr Data Warehouse zugreifen.
Ein Agent fällt vorübergehend aus.	Wenn ein Agent ausfällt, wird der Status seiner Aufgaben in AWS SCT als fehlgeschlagen angezeigt. Wenn Sie warten, kann sich der Agent möglicherweise fangen. In diesem Fall wird der Status seiner Aufgaben in AWS SCT aktualisiert.
Ein Agent fällt dauerhaft aus.	Wenn der Computer, auf dem ein Agent ausgeführt wird, dauerhaft ausfällt und der Agent gerade eine Aufgabe ausführt, können Sie den Agenten austauschen und den neuen Agenten mit der Aufgabe betrauen. Dies ist nur möglich, wenn das Arbeitsverzeichnis des ursprünglichen Agenten sich nicht auf demselben Computer wie der ursprüngliche Agent befindet. Gehen Sie wie folgt vor, um einen Agenten auszutauschen:

Problem	Vorschläge für die Fehlerbehebung
	<ul style="list-style-type: none">• Installieren Sie einen Agenten auf einem neuen Computer.• Konfigurieren Sie den neuen Agenten mit den gleichen Einstellungen wie Portnummer und Arbeitsverzeichnis wie den ursprünglichen Agenten.• Starten Sie den Agenten. Nachdem Start erkennt die Aufgabe den neuen verfügbaren Agenten und fährt mit dem neuen Agenten fort.

Anwendungs-SQL konvertieren mitAWS SCT

Wenn Sie Ihr Datenbankschema von einer Engine in eine andere konvertieren, müssen Sie auch den SQL-Code in Ihren Anwendungen aktualisieren, damit diese mit der neuen Datenbank-Engine anstelle der alten interagieren. Sie können den konvertierten SQL-Code anzeigen, analysieren, bearbeiten und speichern.

Sie können AWS Schema Conversion Tool (AWS SCT) verwenden, um den SQL-Code in C++-, C#, Java- oder anderen Anwendungscode zu konvertieren. Für eine Konvertierung von Oracle nach PostgreSQL können Sie verwendenAWS SCTum SQL*Plus-Code in PSQL zu konvertieren. Für eine Konvertierung von Oracle nach PostgreSQL können Sie auch verwendenAWS SCTum in C#, C++-, Java- und Pro*C-Anwendungen eingebetteten SQL-Code zu konvertieren.

Themen

- [Überblick über die Konvertierung von Anwendungs-SQL](#)
- [Konvertieren von SQL-Code in Ihren Anwendungen mitAWS SCT](#)
- [Konvertieren von SQL-Code in C#-Anwendungen mitAWS SCT](#)
- [Konvertieren von SQL-Code in C++-Anwendungen mitAWS SCT](#)
- [Konvertieren von SQL-Code in Java-Anwendungen mitAWS SCT](#)
- [Konvertierung von SQL-Code in Pro*C-Anwendungen mitAWS SCT](#)

Überblick über die Konvertierung von Anwendungs-SQL

Um den SQL-Code in der Anwendung zu konvertieren, führen Sie die folgenden allgemeinen Schritte aus:

- Erstellen Sie ein Projekt zur Anwendungskonvertierung— Das Projekt zur Anwendungskonvertierung ist dem Projekt zur Konvertierung des Datenbankschemas untergeordnet. Jedes Datenbankschema-Konvertierungsprojekt kann mindestens ein untergeordnetes Anwendungskonvertierungsprojekt enthalten. Weitere Informationen finden Sie unter [Generische Projekte zur Konvertierung von Anwendungen erstellen inAWS SCT](#).
- Analysieren und konvertieren Sie Ihren SQL-Code—AWS SCTanalysiert Ihre Anwendung, extrahiert den SQL-Code und erstellt eine lokale Version der konvertierten SQL, die Sie überprüfen und bearbeiten können. Das Tool ändert den Code in Ihrer Anwendung erst dann, wenn Sie bereit

sind. Weitere Informationen finden Sie unter [Analysieren und konvertieren Sie Ihren SQL-Code inAWS SCT](#).

- Erstellen Sie einen Bericht zur Anwendungsbeurteilung— Der Anwendungsbewertungsbericht enthält wichtige Informationen zur Konvertierung des Anwendungs-SQL-Codes von Ihrem Quelldatenbankschema in Ihr Zieldatenbankschema. Weitere Informationen finden Sie unter [Erstellen und Verwenden desAWS SCTBewertungsbericht inAWS SCT](#).
- Bearbeiten Sie Ihren konvertierten SQL-Code, wenden Sie Änderungen an und speichern Sie ihn— Der Bewertungsbericht enthält eine Liste von SQL-Codeelementen, die nicht automatisch konvertiert werden können. Für diese Elemente können Sie den SQL-Code manuell bearbeiten, um die Konvertierung durchzuführen. Weitere Informationen finden Sie unter [Bearbeiten und speichern Sie Ihren konvertierten SQL-Code mitAWS SCT](#).

Konvertieren von SQL-Code in Ihren Anwendungen mitAWS SCT

VerwendenAWS SCTum in Ihre Anwendungen eingebetteten SQL-Code zu konvertieren. Das GenerikumAWS SCTDer Anwendungskonverter behandelt Ihren Anwendungscode als einfachen Text. Es scannt Ihren Anwendungscode und extrahiert SQL-Code mit regulären Ausdrücken. Dieser Konverter unterstützt verschiedene Arten von Quellcodedateien und arbeitet mit Anwendungscode, der in einer beliebigen Programmiersprache geschrieben ist.

Der generische Anwendungskonverter hat die folgenden Einschränkungen. Er befasst sich nicht eingehend mit der Anwendungslogik, die für die Programmiersprache Ihrer Anwendung spezifisch ist. Außerdem unterstützt der generische Konverter keine SQL-Anweisungen von verschiedenen Anwendungsobjekten wie Funktionen, Parametern, lokalen Variablen usw.

Verwenden Sie sprachspezifische Anwendungs-SQL-Code-Konverter, um die Konvertierung von Anwendungs-SQL-Code zu verbessern. Weitere Informationen finden Sie unter [Konvertieren von SQL-Code in C#-Anwendungen](#), [Umwandeln von SQL-Code in Java-Anwendungen](#) und [Konvertieren von SQL-Code in Pro*C-Anwendungen](#).

Generische Projekte zur Konvertierung von Anwendungen erstellen inAWS SCT

In AWS Schema Conversion Tool ist das Anwendungskonvertierungsprojekt ein untergeordnetes Projekt des Datenbankschema-Konvertierungsprojekts. Jedes Datenbankschema-Konvertierungsprojekt kann mindestens ein untergeordnetes Anwendungskonvertierungsprojekt enthalten.

Note

AWS SCT unterstützt keine Konvertierung zwischen den folgenden Quellen und Zielen:

- Oracle nach Oracle
- PostgreSQL auf PostgreSQL oder Aurora PostgreSQL
- MySQL nach MySQL
- Von SQL Server zu SQL Server
- Von Amazon Redshift auf Amazon Redshift
- SQL Server zu Babelfish
- SQL Server Integration Services auf AWS Glue
- Apache Cassandra nach Amazon DynamoDB

Gehen Sie wie folgt vor, um ein Konvertierungsprojekt für generische Anwendungen zu erstellen.

So erstellen Sie ein Anwendungskonvertierungsprojekt

1. In der AWS Schema Conversion Tool, wähle **Neue generische Anwendung** auf dem **Bewerbungen** Menü

Das Dialogfeld **New application conversion project** wird angezeigt.

Creating a generic application conversion project

Enter the name, location and type of the new application conversion project.

Name: Application conversion project 1

Location: C:\AWS-SCT-Demo Browse

Language: Java Target parameter style: Same as in source

Settings

Don't cast bind variables to SQL types i

Keep object names i

Choose the source database schema that your application uses which is mapped with the target tree object:

- ▼ Schemas [58]
 - ANONYMOUS
 - APPQOSSYS
 - AUDSYS
 - CHINOOK**
 - CTXSYS
 - DVSYs

OK Cancel

2. Fügen Sie die folgenden Projektinformationen hinzu.

Für diesen Parameter	Vorgehensweise
Name	Geben Sie einen Namen für Ihr Anwendungskonvertierungsprojekt ein. Jedes Datenbankschema-Konvertierungsprojekt kann mindestens ein untergeordnetes Anwendungskonvertierungsprojekt aufweisen. Wählen Sie also einen Namen, der sinnvoll ist, wenn Sie zu einem späteren Zeitpunkt weitere Projekte hinzufügen.
Ort	Geben Sie den Speicherort des Quellcodes für Ihre Anwendung ein.
Sprache	Wählen Sie eine der folgenden Optionen aus:

Für diesen Parameter	Vorgehensweise
	<ul style="list-style-type: none"> • Java • C++ • C# • Alle
Zielparameterstil	<p>Wählen Sie die Syntax, die für Bind-Variablen im konvertierten Code verwendet werden soll. Verschiedene Datenbankplattformen verwenden unterschiedliche Syntax für Bind-Variablen. Wählen Sie eine der folgenden Optionen:</p> <ul style="list-style-type: none"> • Same as in source • Positional (?) • Indexed (:1) • Indexed (\$1) • Named (@name) • Named (:name) • Benannt (&Name) • Benannt (\$name) • Benannt (#name) • Benannt (! Name!)
Wählen Sie das Quelldatenbankschema	Wählen Sie im Quellbaum das Schema aus, das Ihre Anwendung verwendet. Stellen Sie sicher, dass dieses Schema Teil einer Zuordnungsregel ist.

3. Wählen Sie Bind-Variablen nicht in SQL-Typen umzuwandeln, um die Konvertierung von Bind-Variablentypen in SQL-Typen zu vermeiden. Diese Option ist nur für eine Konvertierung von Oracle auf PostgreSQL verfügbar.

Ihr Quellanwendungscode enthält beispielsweise die folgende Oracle-Abfrage:

```
SELECT * FROM ACCOUNT WHERE id = ?
```

Wenn Sie wählen **Wandeln Sie Bind-Variablen nicht in SQL-Typen um**, AWS SCT diese Abfrage wird wie folgt konvertiert.

```
SELECT * FROM account WHERE id = ?
```

Wenn Sie löschen **Wandeln Sie Bind-Variablen nicht in SQL-Typen um**, AWS SCT ändert den Typ der Bind-Variablen in `NUMERIC` Datentyp. Das Umrechnungsergebnis wird im Folgenden dargestellt.

```
SELECT * FROM account WHERE id = (?)::NUMERIC
```

4. Wählen Sie **Objektnamen beibehalten** um zu vermeiden, dass der Schemaname zum Namen des konvertierten Objekts hinzugefügt wird. Diese Option ist nur für eine Konvertierung von Oracle auf PostgreSQL verfügbar.

Nehmen Sie zum Beispiel an, dass der Quellanwendungscode die folgende Oracle-Abfrage enthält.

```
SELECT * FROM ACCOUNT
```

Wenn du auswählst **Objektnamen beibehalten**, AWS SCT diese Abfrage wird wie folgt konvertiert.

```
SELECT * FROM account
```

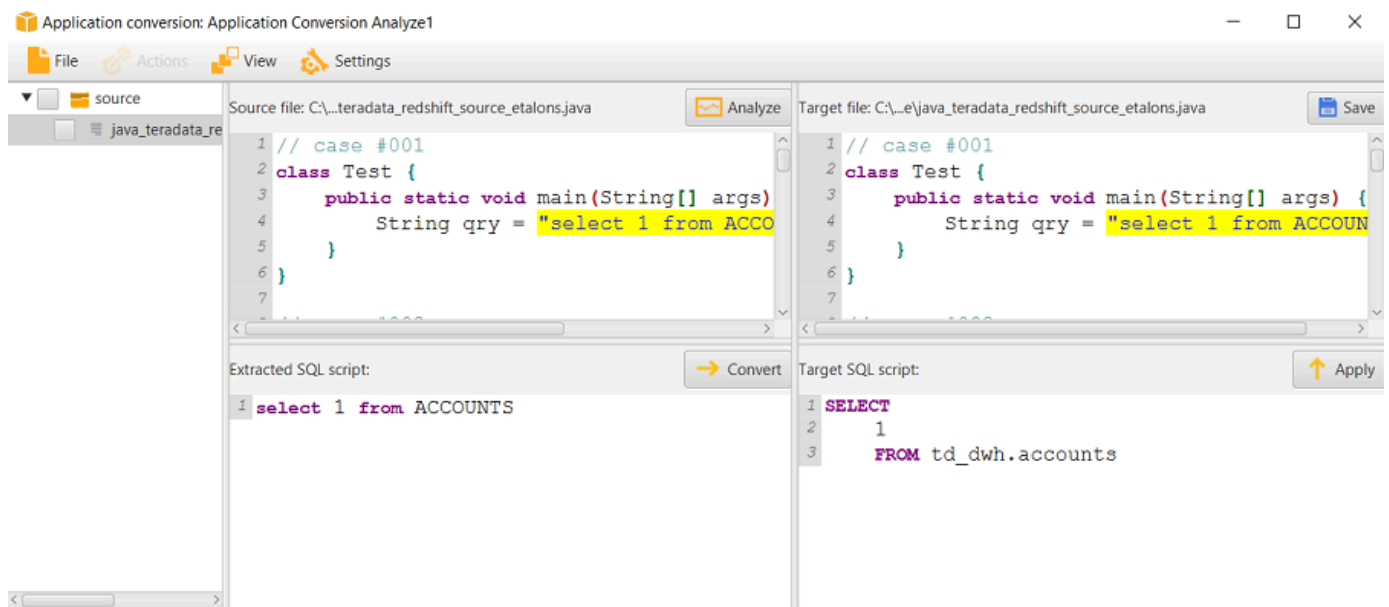
Wenn Sie löschen **Objektnamen beibehalten**, AWS SCT fügt den Schemanamen zum Namen der Tabelle hinzu. Das Umrechnungsergebnis wird im Folgenden dargestellt.

```
SELECT * FROM schema_name.account
```

Wenn Ihr Quellcode die Namen der übergeordneten Objekte in den Namen der Objekte enthält, AWS SCT verwendet dieses Format im konvertierten Code. Ignorieren Sie in diesem Fall die **Objektnamen beibehalten** Option weil AWS SCT fügt die Namen der übergeordneten Objekte in den konvertierten Code ein.

5. Klicken Sie auf **OK**, um Ihr Anwendungskonvertierungsprojekt zu erstellen.

Das Projektfenster wird geöffnet.



Verwaltung von Projekten zur Anwendungskonvertierung in AWS SCT

Sie können ein vorhandenes Projekt zur Anwendungskonvertierung öffnen und mehrere Projekte zur Anwendungskonvertierung hinzufügen.

Nachdem Sie ein Projekt zur Anwendungskonvertierung erstellt haben, wird das Projektfenster automatisch geöffnet. Sie können das Projektfenster für die Anwendungskonvertierung schließen und später wieder darauf zugreifen.

Um ein vorhandenes Projekt zur Anwendungskonvertierung zu öffnen

1. Wählen Sie im linken Bereich aus, klicken Sie auf den Projektknoten Anwendungskonvertierung und öffnen Sie das Kontextmenü (rechte Maustaste).
2. Wählen Sie Anwendung verwalten.

Um ein zusätzliches Projekt zur Anwendungskonvertierung hinzuzufügen

1. Wählen Sie im linken Bereich aus, klicken Sie auf den Projektknoten Anwendungskonvertierung und öffnen Sie das Kontextmenü (rechte Maustaste).
2. Wählen Sie New application (Neue Anwendung) aus.
3. Geben Sie die Informationen ein, die für die Erstellung eines neuen Anwendungskonvertierungsprojekts erforderlich sind. Weitere Informationen finden Sie unter [Projekte zur Konvertierung generischer Anwendungen erstellen](#).

Analysieren und konvertieren Sie Ihren SQL-Code in AWS SCT

Verwenden Sie die folgende Vorgehensweise, um Ihren SQL-Code in der zu analysieren und zu konvertieren. AWS Schema Conversion Tool.

So analysieren und konvertieren Sie den SQL-Code

- Öffnen Sie ein vorhandenes Projekt zur Anwendungskonvertierung und wählen Sie **Analysieren**.
AWS SCT analysiert Ihren Anwendungscode und extrahiert den SQL-Code. AWS SCT zeigt den extrahierten SQL-Code in der **Analysierte SQL-Skripten** Liste.
- Für **Analysierte SQL-Skripten**, wählen Sie ein Element aus, um den extrahierten SQL-Code zu überprüfen. AWS SCT zeigt den Code des ausgewählten Elements in der **Extrahiertes SQL-Skript** Fenstergröße.
- Wählen **Konvertieren** um den SQL-Code zu konvertieren. **Extrahiertes SQL-Skript** Fenstergröße AWS SCT konvertiert den Code in ein Format, das mit Ihrer Zieldatenbank kompatibel ist.

Sie können den konvertierten SQL-Code bearbeiten. Weitere Informationen finden Sie unter [Bearbeiten und Speichern Ihres konvertierten SQL-Codes](#).

The screenshot displays the AWS SCT interface with the following components:

- Source file:** C:\..._redshift_source_etalons.java.old. The code snippet shows a Java string variable 'qry' containing a SQL query: `String qry = "select 1 a from ACCOUNTS b";`
- Target file:** C:\..._redshift_source_etalons.java. The converted SQL code is: `SELECT 1 AS a FROM td_dwh.accounts AS b`
- Extracted SQL script:** `1 select 1 a from ACCOUNTS b`
- Target SQL script:** `1 SELECT 2 1 AS a 3 FROM td_dwh.accounts AS b`
- Parsed SQL scripts table:**

Source file	Position	Extracted code	Converted code
C:\And...s.java	Line 11 22:50	✓ "select 1 a from ACCOUNTS b"	✓ SELECT 1 AS a FROM td_dwh.accounts AS b
C:\And...s.java	Line 18 22:46	✓ "select * from ACCOUNTS"	
C:\And...s.java	Line 25 22:50	✓ "select a.* from ACCOUNTS a"	

- Wenn Sie einen Bewertungsbericht zur Anwendungskonvertierung erstellen, AWS SCT konvertiert alle extrahierten SQL-Codeelemente. Weitere Informationen finden Sie unter [Erstellung und Verwendung des Bewertungsberichts](#).

Erstellen und Verwenden des AWS SCT Bewertungsberichts in AWS SCT

Der Bewertungsbericht zur Konvertierung von Anwendungen enthält Informationen zur Konvertierung des Anwendungs-SQL-Codes in ein Format, das mit Ihrer Zieldatenbank kompatibel ist. Der Bericht beschreibt den gesamten extrahierten SQL-Code, den gesamten konvertierten SQL-Code und die Aktionspunkte für den folgenden SQL-Code, den AWS SCT nicht konvertieren kann.

Erstellen eines Bewertungsberichts zur Anwendungskonvertierung

Gehen Sie wie folgt vor, um einen Bewertungsbericht zur Anwendungskonvertierung zu erstellen.

Um einen Bewertungsbericht zur Anwendungskonvertierung zu erstellen

1. Wählen Sie im Projektfenster für die Anwendungskonvertierung Bericht erstellen auf dem Aktionen-Menü

AWS SCT erstellt den Bewertungsbericht zur Anwendungskonvertierung und öffnet ihn im Projektfenster für die Anwendungskonvertierung.

2. Prüfen Sie die Registerkarte Summary.

Der Zusammenfassungsauf der Registerkarte (unten) werden die zusammenfassenden Informationen aus dem Bericht zur Anwendungsbeurteilung angezeigt. Es werden sowohl die automatisch konvertierten SQL-Codeelemente als auch Elemente, die nicht automatisch konvertiert wurden, angezeigt.



3. Wählen Sie Aktionen zur SQL-Extraktion.

Sehen Sie sich die Liste der SQL-Codeelemente an, die AWS SCT nicht aus Ihrem Quellcode extrahieren kann.

4. Wählen Sie Aktionen zur SQL-Konvertierung.

Sehen Sie sich die Liste der SQL-Codeelemente an, die AWS SCT nicht automatisch konvertieren kann. Verwenden Sie die empfohlenen Aktionen, um den SQL-Code manuell zu konvertieren. Weitere Informationen zur Bearbeitung Ihres konvertierten SQL-Codes finden Sie unter [Bearbeiten und speichern Sie Ihren konvertierten SQL-Code mit AWS SCT](#).

5. (Optional) Speichern Sie eine lokale Kopie des Berichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

- Wählen Sie **Speichern** oben rechts, um den Bericht als PDF-Datei zu speichern.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Anwendungskonvertierung.

- Wählen Sie **Speichern der Ordnergröße** oben rechts, um den Bericht als CSV-Datei zu speichern.

Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung des SQL-Codes erforderlich ist.

Bearbeiten und speichern Sie Ihren konvertierten SQL-Code mit AWS SCT

Der Bewertungsbericht enthält eine Liste mit SQL-Codeelementen, die AWS SCT nicht konvertieren kann. Für jeden Artikel erstellt AWS SCT einen Aktionspunkt auf dem **Aktionen zur SQL-Konvertierung** Tab. Für diese Elemente können Sie den SQL-Code manuell bearbeiten, um die Konvertierung durchzuführen.

Führen Sie die folgenden Schritte aus, um den konvertierten SQL-Code zu bearbeiten, die Änderungen zu übernehmen und sie anschließend zu speichern.

So können Sie konvertierten SQL-Code bearbeiten, Änderungen daran übernehmen und ihn speichern

1. Bearbeiten Sie den konvertierten SQL-Code direkt im Bereich **Target SQL script**. Wenn kein konvertierter Code angezeigt wird, können Sie auf den Bereich klicken und mit der Eingabe beginnen.
2. Nachdem Sie die Bearbeitung des konvertierten SQL-Codes abgeschlossen haben, wählen Sie **Apply** aus. An diesem Punkt werden die Änderungen im Arbeitsspeicher gespeichert, aber noch nicht in Ihre Datei geschrieben.
3. Wählen Sie **Save** aus, um Ihre Änderungen in Ihrer Datei zu speichern.

Wenn Sie **Speichern** wählen, überschreiben Sie Ihre Originaldatei. Erstellen Sie vor dem Speichern eine Kopie der ursprünglichen Datei, sodass Sie einen Datensatz des ursprünglichen Anwendungscodes haben.

Konvertieren von SQL-Code in C#-Anwendungen mit AWS SCT

Für eine Konvertierung von Oracle nach PostgreSQL können Sie verwenden AWS Schema Conversion Tool (AWS SCT), um in Ihre C#-Anwendungen eingebetteten SQL-Code zu konvertieren. Dieser spezielle C#-Anwendungskonverter versteht die Anwendungslogik. Er sammelt Anweisungen, die sich in verschiedenen Anwendungsobjekten befinden, z. B. Funktionen, Parameter, lokale Variablen usw.

Aufgrund dieser gründlichen Analyse liefert der SQL-Code-Konverter für C#-Anwendungen bessere Konvertierungsergebnisse als der generische Konverter.

Projekte zur Konvertierung von C#-Anwendungen erstellen in AWS SCT

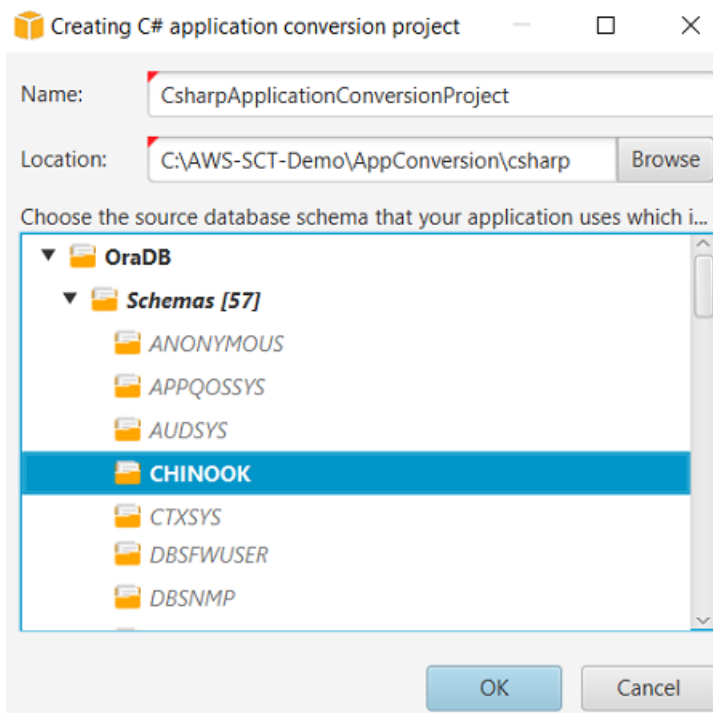
Sie können ein C#-Anwendungskonvertierungsprojekt nur für die Konvertierung von Oracle-Datenbankschemas in PostgreSQL-Datenbankschemas erstellen. Stellen Sie sicher, dass Sie Ihrem Projekt eine Zuordnungsregel hinzufügen, die ein Oracle-Quellschema und eine PostgreSQL-Zieldatenbank enthält. Weitere Informationen finden Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#).

Sie können mehrere Projekte zur Anwendungskonvertierung zu einem einzigen hinzufügen AWS SCT Projekt. Gehen Sie wie folgt vor, um ein Projekt zur Konvertierung von C#-Anwendungen zu erstellen.

So erstellen Sie ein Projekt zur Konvertierung von C#-Anwendungen

1. Erstellen Sie ein Datenbankkonvertierungsprojekt und fügen Sie eine Oracle-Quelldatenbank hinzu. Weitere Informationen erhalten Sie unter [Ein AWS SCT Projekt erstellen](#) und [Hinzufügen von Datenbankservern zu einem AWS SCT Projekt](#).
2. Fügen Sie eine Zuordnungsregel hinzu, die Ihre Oracle-Quelldatenbank und eine PostgreSQL-Zieldatenbank umfasst. Sie können eine PostgreSQL-Zieldatenbank hinzufügen oder eine virtuelle PostgreSQL-Zieldatenbankplattform in einer Zuordnungsregel verwenden. Weitere Informationen erhalten Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#) und [Verwendung virtueller Ziele](#).
3. Auf der Ansicht Menü, wählen Hauptgröße.
4. Auf der Bewerbungen Menü, wählen Neue C#-Anwendung.

Das Ein Projekt zur Konvertierung von C#-Anwendungen erstellen ein Dialogfeld wird angezeigt.



5. Für **Name**, geben Sie einen Namen für Ihr C#-Anwendungskonvertierungsprojekt ein. Da jedes Datenbankschema-Konvertierungsprojekt ein oder mehrere untergeordnete Anwendungskonvertierungsprojekte haben kann, wählen Sie einen Namen, der sinnvoll ist, wenn Sie mehrere Projekte hinzufügen.
6. Für **Ort**, geben Sie die Position des Quellcodes Ihrer Anwendung ein.
7. Wählen Sie im Quellbaum das Schema aus, das Ihre Anwendung verwendet. Stellen Sie sicher, dass dieses Schema Teil einer Zuordnungsregel ist. AWS SCT hebt die Schemas, die Teil einer Zuordnungsregel sind, fett hervor.
8. Wählen Sie **OK** um Ihr C#-Anwendungskonvertierungsprojekt zu erstellen.
9. Finden Sie Ihr Projekt zur Konvertierung von C#-Anwendungen im **AnwendungenKnoten** im linken Bereich.

Konvertieren Sie den SQL-Code Ihrer C#-Anwendung in AWS SCT

Nachdem Sie Ihre C#-Anwendung zum hinzugefügt haben AWS SCT Projekt, konvertieren Sie den SQL-Code aus dieser Anwendung in ein Format, das mit Ihrer Zieldatenbankplattform kompatibel ist. Verwenden Sie das folgende Verfahren, um den in Ihre C#-Anwendung eingebetteten SQL-Code in der AWS Schema Conversion Tool.

Um Ihren SQL-Code zu konvertieren

1. Erweitern Sie das C#-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie konvertieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen Konvertiere. AWS SCT analysiert Ihre Quellcode-Dateien, bestimmt die Anwendungslogik und lädt Code-Metadaten in das Projekt. Diese Code-Metadaten umfassen C#-Klassen, Objekte, Methoden, globale Variablen, Schnittstellen usw.

Im Zieldatenbank AWS SCT erstellt eine ähnliche Ordnerstruktur wie Ihr Quellanwendungsprojekt. Hier können Sie den konvertierten Anwendungscode überprüfen.

```
Source Oracle file: SpecialEscapeSequences.cs
Properties Text Related converted objects Statistics Settings
18      {
19          string strSql = "SELECT *\n" +
20                          "FROM\t JAVADB.DATETYPE_MIXED_ALL\n\x" +
21                          "WHERE COL_CHAR = \ 'CHAR\ '";
22
23          command.CommandText = strSql;
24          command.ExecuteNonQuery();
25      }
26      connection.Close();
Cursor position: 1005

Target Amazon RDS for PostgreSQL file: SpecialEscapeSequences.cs
Properties Text Apply status Key management
19      {
20          string strSql = "SELECT *\n" +
21                          "FROM\t JAVADB.DATETYPE_MIXED_ALL\n\x" +
22                          "WHERE COL_CHAR = \ 'CHAR\ '";
23
24          command.CommandText = strSql;
25          command.ExecuteNonQuery();
26      }
27      connection.Close();
```

4. Speichern Sie Ihren konvertierten Anwendungscode. Weitere Informationen finden Sie unter [Speichern Sie Ihren konvertierten Anwendungscode](#).

Ihre C#-Anwendungen können SQL-Code enthalten, der mit verschiedenen Quelldatenbanken interagiert. Sie können mehrere dieser Quelldatenbanken zu PostgreSQL migrieren. Stellen Sie in diesem Fall sicher, dass Sie keinen SQL-Code konvertieren, der mit Datenbanken interagiert, die Sie aus dem Migrationsbereich ausgeschlossen haben. Sie können Quelldateien Ihrer C#-Anwendung vom Konvertierungsbereich ausschließen. Deaktivieren Sie dazu die Kontrollkästchen für die Namen der Dateien, die Sie aus dem Konvertierungsbereich ausschließen möchten.

Nachdem Sie den Konvertierungsumfang geändert haben, analysiert AWS SCT weiterhin den SQL-Code aller Quelldateien Ihrer C#-Anwendungen. Dann kopiert AWS SCT alle Quelldateien, die Sie aus dem Konvertierungsbereich ausgeschlossen haben, in den Zielordner. Dieser Vorgang ermöglicht es, Ihre Anwendung zu erstellen, nachdem Sie Ihre konvertierten Anwendungsdateien gespeichert haben.

Speichern Sie Ihren konvertierten Anwendungscode mit AWS SCT

Mit dem folgenden Verfahren können Sie Ihren konvertierten Anwendungscode speichern.

Um Ihren konvertierten Anwendungscode zu speichern

1. Erweitern Sie das C#-Knoten unter Anwendungen in der Zieldatenbank.
2. Wählen Sie Ihre konvertierte Anwendung und wählen Sie Speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem der konvertierte Anwendungscode gespeichert werden soll, und wählen Sie Wählen eines Ordners.

Verwaltung von Projekten zur Konvertierung von C#-Anwendungen in AWS SCT

Sie können mehrere Projekte zur Konvertierung von C#-Anwendungen hinzufügen und den Anwendungscode in der Datei aktualisieren. Sie können ein C#-Konvertierungsprojekt aus Ihrem AWS SCT-Projekt entfernen.

Um ein zusätzliches Projekt zur Konvertierung von C#-Anwendungen hinzuzufügen

1. Erweitern Sie das Anwendungen-Knoten im linken Bereich.
2. Wählen Sie die C#-Node, und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie New application (Neue Anwendung) aus.
4. Geben Sie die Informationen ein, die für die Erstellung eines neuen Projekts zur Konvertierung von C#-Anwendungen erforderlich sind. Weitere Informationen finden Sie unter [Projekte zur Konvertierung von C#-Anwendungen erstellen](#).

Nachdem Sie Änderungen an Ihrem Quellanwendungscode vorgenommen haben, laden Sie ihn in den AWS SCT-Projekt

Um den aktualisierten Anwendungscode hochzuladen

1. Erweitern Sie das **C#**-Knoten unter **Anwendungen** im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie aktualisieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen **Auffrischen** und dann wähle **Ja**.

AWS SCT lädt Ihren Anwendungscode aus den Quelldateien hoch und entfernt die Konvertierungsergebnisse. Um Codeänderungen beizubehalten, die Sie in vorgenommen haben, erstellen Sie zusammen mit den Konvertierungsergebnissen ein neues **C#**-Konvertierungsprojekt.

Um ein **C#**-Anwendungskonvertierungsprojekt zu entfernen

1. Erweitern Sie das **C#**-Knoten unter **Anwendungen** im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie entfernen möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen **Löschen** und dann wähle **OKAY**.

Erstellen eines Bewertungsberichts zur **C#**-Anwendungskonvertierung in AWS SCT

Der Bewertungsbericht zur Konvertierung von **C#**-Anwendungen enthält Informationen zur Konvertierung des in Ihrer **C#**-Anwendung eingebetteten SQL-Codes in ein Format, das mit Ihrer Zieldatenbank kompatibel ist. Der Bewertungsbericht enthält Konvertierungsdetails für alle SQL-Ausführungspunkte und alle Quellcodedateien. Der Bewertungsbericht enthält auch folgende Aktionspunkte für SQL-Code, den AWS SCT nicht konvertieren kann.

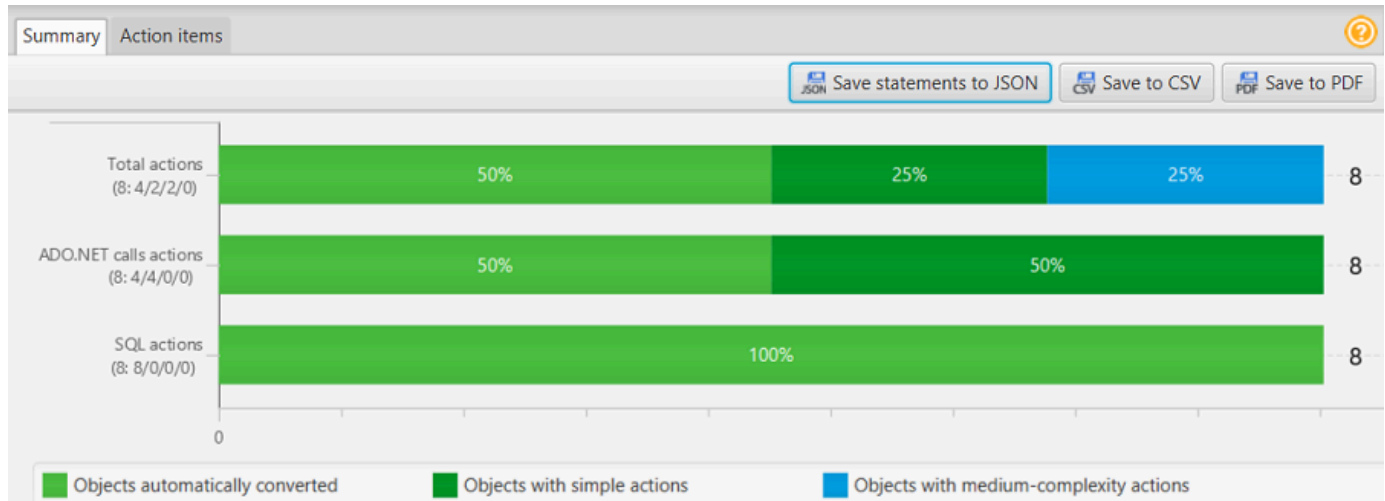
Gehen Sie wie folgt vor, um einen Bewertungsbericht zur **C#**-Anwendungskonvertierung zu erstellen.

So erstellen Sie einen Bewertungsbericht zur **C#**-Anwendungskonvertierung

1. Erweitern Sie **C#**-Knoten unter **Anwendungen** im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie konvertieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen **Konvertiere**.

4. Auf der AnsichtMenü, wählen Ansicht des Bewertungsberichts.
5. Sehen der Zusammenfassung Tab

Der Zusammenfassung Auf der Registerkarte (unten) werden die zusammenfassenden Informationen aus dem Bericht zur Bewertung der C#-Anwendung angezeigt. Es zeigt die Konvertierungsergebnisse für alle SQL-Ausführungspunkte und alle Quellcodedateien.



6. Wählen Sie Speichern Sie Anweisungen in JSON um den extrahierten SQL-Code aus Ihrer C#-Anwendung als JSON-Datei zu speichern.
7. (Optional) Speichern Sie eine lokale Kopie des Berichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):
 - Wählen Speichern oben rechts, um den Bericht als PDF-Datei zu speichern.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Anwendungskonvertierung.

- Wählen Speichern der Ordnergröße oben rechts, um den Bericht als CSV-Datei zu speichern.

Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung des SQL-Codes erforderlich ist.

Konvertieren von SQL-Code in C++-Anwendungen mit AWS SCT

Für eine Konvertierung von Oracle nach PostgreSQL können Sie verwenden AWS SCT um in Ihre C++-Anwendungen eingebetteten SQL-Code zu konvertieren. Dieser spezielle C++-

Anwendungskonverter versteht die Anwendungslogik. Er sammelt Anweisungen, die sich in verschiedenen Anwendungsobjekten befinden, wie Funktionen, Parameter, lokale Variablen usw.

Aufgrund dieser gründlichen Analyse liefert der SQL-Code-Konverter für C++-Anwendungen bessere Konvertierungsergebnisse als der generische Konverter.

Projekte zur Konvertierung von C++-Anwendungen erstellen in AWS SCT

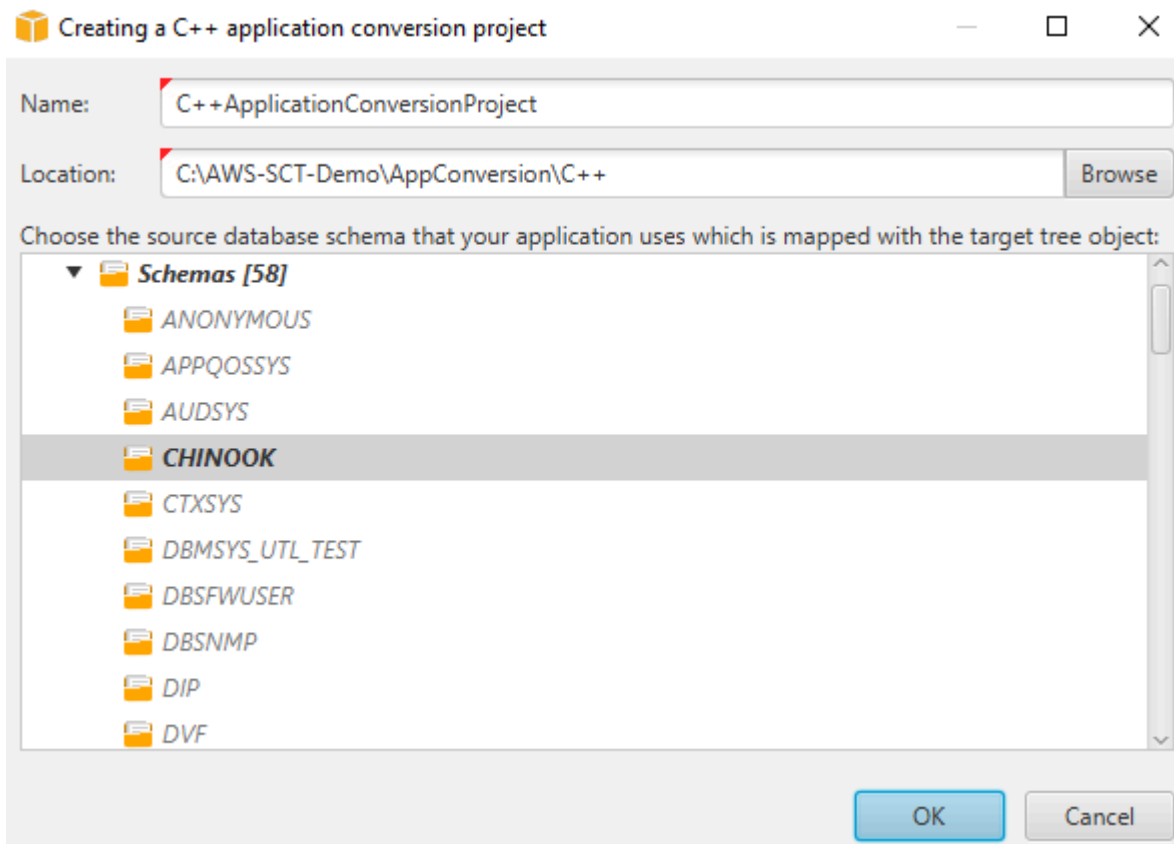
Sie können ein C++-Anwendungskonvertierungsprojekt nur für die Konvertierung von Oracle-Datenbankschemas in PostgreSQL-Datenbankschemas erstellen. Stellen Sie sicher, dass Sie Ihrem Projekt eine Zuordnungsregel hinzufügen, die ein Oracle-Quellschema und eine PostgreSQL-Zieldatenbank enthält. Weitere Informationen finden Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#).

Sie können mehrere Projekte zur Anwendungskonvertierung zu einem einzigen hinzufügen AWS SCT-Projekt

Um ein Projekt zur Konvertierung von C++-Anwendungen zu erstellen

1. Erstellen Sie ein Datenbankkonvertierungsprojekt und fügen Sie eine Oracle-Quelldatenbank hinzu. Weitere Informationen erhalten Sie unter [Ein AWS SCT Projekt erstellen](#) und [Hinzufügen von Datenbankservern zu einem AWS SCT Projekt](#).
2. Fügen Sie eine Zuordnungsregel hinzu, die Ihre Oracle-Quelldatenbank und eine PostgreSQL-Zieldatenbank umfasst. Sie können eine PostgreSQL-Zieldatenbank hinzufügen oder eine virtuelle PostgreSQL-Zieldatenbankplattform in einer Zuordnungsregel verwenden. Weitere Informationen erhalten Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#) und [Verwendung virtueller Ziele](#).
3. Auf dem Ansicht-Menü, wählen Hauptgröße.
4. Auf dem Bewerbungen-Menü, wählen Neue C++-Anwendung.

Das Ein Projekt zur Konvertierung von C++-Anwendungen erstellen Dialogfeld wird angezeigt.



5. Für `Name`, geben Sie einen Namen für Ihr C++-Anwendungskonvertierungsprojekt ein. Da jedes Datenbankschema-Konvertierungsprojekt ein oder mehrere untergeordnete Anwendungskonvertierungsprojekte haben kann, wählen Sie einen Namen, der Sinn macht, wenn Sie mehrere Projekte hinzufügen.
6. Für `Ort`, geben Sie die Position des Quellcodes Ihrer Anwendung ein.
7. Wählen Sie im Quellbaum das Schema aus, das Ihre Anwendung verwendet. Stellen Sie sicher, dass dieses Schema Teil einer Zuordnungsregel ist. AWS SCT hebt die Schemas, die Teil einer Zuordnungsregel sind, fett hervor.
8. Wählen `OK` um Ihr C++-Anwendungskonvertierungsprojekt zu erstellen.
9. Finden Sie Ihr Projekt zur Konvertierung von C++-Anwendungen im `Anwendungen`-Knoten im linken Bereich.

Konvertieren Sie den SQL-Code Ihrer C++-Anwendung in AWS SCT

Nachdem Sie Ihre C++-Anwendung zum hinzugefügt haben AWS SCT Projekt, konvertieren Sie den SQL-Code aus dieser Anwendung in ein Format, das mit Ihrer Zieldatenbankplattform kompatibel ist.

Verwenden Sie das folgende Verfahren, um den in Ihre C++-Anwendung eingebetteten SQL-Code zu analysieren und zu konvertieren in AWS SCT.

Um Ihren SQL-Code zu konvertieren

1. Erweitern Sie das C++-Knoten unter Anwendungen im linken Bereich und wählen Sie die zu konvertierende Anwendung aus.
2. In der Quellprojekt der Oracle-Anwendung, wählen Einstellungen. Überprüfen und bearbeiten Sie die Konvertierungseinstellungen für die ausgewählte C++-Anwendung. Sie können auch die Konvertierungseinstellungen für alle C++-Anwendungen angeben, die Sie zu Ihrer AWS SCT-Projekt Weitere Informationen finden Sie unter [Verwaltung von Projekten zur Konvertierung von C++-Anwendungen](#).
3. Für Compiler-Typ, wählen Sie den Compiler, den Sie für den Quellcode Ihrer C++-Anwendung verwenden. AWS SCT unterstützt die folgenden C++-Compiler: Microsoft Visual C++, GCC, die GNU-Compiler-Sammlung, und Clang. Die Standardoption ist Microsoft Visual C++.
4. Für Benutzerdefinierte Makros, geben Sie den Pfad zu der Datei ein, die benutzerdefinierte Makros aus Ihrem C++-Projekt enthält. Stellen Sie sicher, dass diese Datei die folgende Struktur hat: `#define name value`. Im vorherigen Beispiel `value` ist ein optionaler Parameter. Der Standardwert für diesen optionalen Parameter ist `1`.

Um diese Datei zu erstellen, öffnen Sie Ihr Projekt in Microsoft Visual Studio, und wählen Sie dann Projekt, Eigenschaften, C/C++, und Präprozessor. Für Präprozessor-Definitionen, wählen Bearbeiten und die Namen und Werte in eine neue Textdatei kopieren. Fügen Sie dann für jede Zeichenfolge in der Datei das folgende Präfix hinzu: `#define` .

5. Für Externe Include-Verzeichnisse, geben Sie die Pfade zu den Ordnern ein, die externe Bibliotheken enthalten, die Sie in Ihrem C++-Projekt verwenden.
6. Wählen Sie im linken Bereich die Anwendung aus, die Sie konvertieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
7. Wählen Konvertiere. AWS SCT analysiert Ihre Quellcode-Dateien, bestimmt die Anwendungslogik und lädt Code-Metadaten in das Projekt. Diese Code-Metadaten umfassen C++-Klassen, Objekte, Methoden, globale Variablen, Schnittstellen usw.

Im Zieldatenbank-Panel AWS SCT erstellt eine ähnliche Ordnerstruktur wie Ihr Quellanwendungsprojekt. Hier können Sie den konvertierten Anwendungscode überprüfen, wie folgt.

The screenshot displays two windows from the AWS Schema Conversion Tool. The top window, titled 'Source Oracle file: StringInitialization.cpp', shows C++ code with line numbers 44 to 55. The code includes a conditional execution of an Oracle SQL query: `SELECT JAVADB.GET_INT() FROM DUAL`. The bottom window, titled 'Target Amazon RDS for PostgreSQL file: StringInitialization.cpp', shows the converted C++ code with line numbers 45 to 56. The converted code uses the PostgreSQL equivalent query: `SELECT javadb.get_int()`.

- Speichern Sie Ihren konvertierten Anwendungscode. Weitere Informationen finden Sie unter [Speichern Sie Ihren konvertierten Anwendungscode](#).

Speichern Sie Ihren konvertierten Anwendungscode mit AWS SCT

Mit dem folgenden Verfahren können Sie Ihren konvertierten Anwendungscode speichern.

Um Ihren konvertierten Anwendungscode zu speichern

- Erweitern Sie das C++-Knoten unter Anwendungen im Zieldatenbank-Panel.
- Wählen Sie Ihre konvertierte Anwendung und wählen Sie Speichern.
- Geben Sie den Pfad zu dem Ordner ein, in dem der konvertierte Anwendungscode gespeichert werden soll, und wählen Sie Wählen Sie einen Ordner.

Verwaltung von Projekten zur Konvertierung von C++-Anwendungen in AWS SCT

Sie können mehrere Projekte zur Konvertierung von C++-Anwendungen hinzufügen, die Konvertierungseinstellungen bearbeiten, den C++-Anwendungscode aktualisieren oder ein C++-Konvertierungsprojekt aus Ihrem entfernen AWS SCT Projekt

Um ein zusätzliches Projekt zur Konvertierung von C++-Anwendungen hinzuzufügen

1. Erweitern Sie das Anwendungen Knoten im linken Bereich.
2. Wählen Sie die C++ Node, und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie New application (Neue Anwendung) aus.
4. Geben Sie die Informationen ein, die zum Erstellen eines neuen Projekts zur Konvertierung von C++-Anwendungen erforderlich sind. Weitere Informationen finden Sie unter [Projekte zur Konvertierung von C++-Anwendungen erstellen](#).

Sie können Konvertierungseinstellungen für alle Projekte zur Konvertierung von C++-Anwendungen in Ihrem AWS SCT Projekt

Um die Konvertierungseinstellungen für alle C++-Anwendungen zu bearbeiten

1. Auf dem Einstellungen Menü, wählen Einstellungen des Projekts, und wählen Sie dann Konvertierung von Anwendungen.
2. Für Compiler-Typ, wählen Sie den Compiler, den Sie für den Quellcode Ihrer C++-Anwendung verwenden. AWS SCT unterstützt die folgenden C++-Compiler: Microsoft Visual C++, GCC, die GNU-Compiler-Sammlung, und Clirren. Die Standardoption ist Microsoft Visual C++.
3. Für Benutzerdefinierte Makros, geben Sie den Pfad zu der Datei ein, die benutzerdefinierte Makros aus Ihrem C++-Projekt enthält. Stellen Sie sicher, dass diese Datei die folgende Struktur hat: `#define name value`. Im vorherigen Beispiel `value` ist ein optionaler Parameter. Der Standardwert für diesen optionalen Parameter ist `1`.

Um diese Datei zu erstellen, öffnen Sie Ihr Projekt in Microsoft Visual Studio, und wählen Sie dann Projekt, Eigenschaften, C/C ++, und Präprozessor. Für Präprozessor-Definitionen, wählen Bearbeiten und die Namen und Werte in eine neue Textdatei kopieren. Fügen Sie dann für jede Zeichenfolge in der Datei das folgende Präfix hinzu: `#define` .

4. Für Externe Include-Verzeichnisse, geben Sie die Pfade zu den Ordnern ein, die externe Bibliotheken enthalten, die Sie in Ihrem C++-Projekt verwenden.
5. Wählen OKAY um die Projekteinstellungen zu speichern und das Fenster zu schließen.

Sie können auch Konvertierungseinstellungen für jedes Projekt zur Konvertierung von C++-Anwendungen angeben. Weitere Informationen finden Sie unter [Konvertieren Sie den SQL-Code Ihrer C++-Anwendung](#).

Nachdem Sie Änderungen an Ihrem Quellanwendungscode vorgenommen haben, laden Sie ihn in den AWS SCT Projekt

Um den aktualisierten Anwendungscode hochzuladen

1. Erweitern Sie das C++ Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie aktualisieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen Auffrischen und dann wähle Ja.

AWS SCT lädt Ihren Anwendungscode aus den Quelldateien hoch und entfernt die Konvertierungsergebnisse. Um Codeänderungen beizubehalten, die Sie in vorgenommen haben, erstellen Sie mit den Konvertierungsergebnissen ein neues C++-Konvertierungsprojekt.

Ebenfalls entfernt AWS SCT die Einstellungen für die Anwendungskonvertierung, die Sie für die ausgewählte Anwendung angegeben haben. Nachdem Sie den aktualisierten Anwendungscode hochgeladen haben, wendet AWS SCT die Standardwerte aus den Projekteinstellungen an.

Um ein Projekt zur Konvertierung einer C++-Anwendung zu entfernen

1. Erweitern Sie das C++ Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie entfernen möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen Löschen und dann wähle OKAY.

Erstellen eines Bewertungsberichts zur Konvertierung von C++-Anwendungen in AWS SCT

Der Bewertungsbericht zur Konvertierung von C++-Anwendungen enthält Informationen zur Konvertierung des in Ihrer C++-Anwendung eingebetteten SQL-Codes in ein Format, das mit Ihrer Zieldatenbank kompatibel ist. Der Bewertungsbericht enthält Konvertierungsdetails für alle SQL-Ausführungspunkte und alle Quellcode-Dateien. Der Bewertungsbericht enthält auch Aktionspunkte für SQL-Code, die AWS SCT nicht konvertieren kann.

Um einen Bewertungsbericht zur Konvertierung von C++-Anwendungen zu erstellen

1. Erweitern Sie C++-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie konvertieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen Sie Konvertiere.
4. Auf der Ansicht Menü, wählen Sie Ansicht des Bewertungsberichts.
5. Sehen Sie die Zusammenfassung Tab

Die Zusammenfassung Auf der Registerkarte werden die zusammenfassenden Informationen aus dem Bericht zur Bewertung der C++-Anwendung angezeigt. Es zeigt die Konvertierungsergebnisse für alle SQL-Ausführungspunkte und alle Quellcode-Dateien.

6. Wählen Sie Speichern Sie Anweisungen in JSON um den extrahierten SQL-Code aus Ihrer Java-Anwendung als JSON-Datei zu speichern.
7. (Optional) Speichern Sie eine lokale Kopie des Berichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):
 - Wählen Sie Speichern oben rechts, um den Bericht als PDF-Datei zu speichern.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Anwendungskonvertierung.

- Wählen Sie Speichern der Ordnergröße oben rechts, um den Bericht als CSV-Datei zu speichern.

Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung des SQL-Codes erforderlich ist.

Konvertieren von SQL-Code in Java-Anwendungen mit AWS SCT

Für eine Konvertierung von Oracle nach PostgreSQL können Sie verwenden AWS Schema Conversion Tool um in Ihre Java-Anwendungen eingebetteten SQL-Code zu konvertieren. Dieser spezielle Java-Anwendungskonverter versteht die Anwendungslogik. Er sammelt Anweisungen, die sich in verschiedenen Anwendungsobjekten befinden, wie Funktionen, Parameter, lokale Variablen usw.

Aufgrund dieser gründlichen Analyse liefert der SQL-Code-Konverter für Java-Anwendungen bessere Konvertierungsergebnisse als der generische Konverter.

Wenn Ihre Java-Anwendung das verwendet MyBatis Framework für die Interaktion mit Datenbanken, dann können Sie verwenden AWS SCT um eingebettete SQL-Anweisungen zu konvertieren MyBatis XML-Dateien und Anmerkungen. Um die Logik dieser SQL-Anweisungen zu verstehen, AWS SCT verwendet die MyBatis Konfigurationsdatei. AWS SCT kann diese Datei automatisch in Ihrem Anwendungsordner finden, oder Sie können den Pfad zu dieser Datei manuell eingeben.

Projekte zur Konvertierung von Java-Anwendungen erstellen in AWS SCT

Sie können ein Konvertierungsprojekt für Java-Anwendungen nur für die Konvertierung von Oracle-Datenbankschemas in PostgreSQL-Datenbankschemas erstellen. Stellen Sie sicher, dass Sie Ihrem Projekt eine Zuordnungsregel hinzufügen, die ein Oracle-Quellschema und eine PostgreSQL-Zieldatenbank enthält. Weitere Informationen finden Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#).

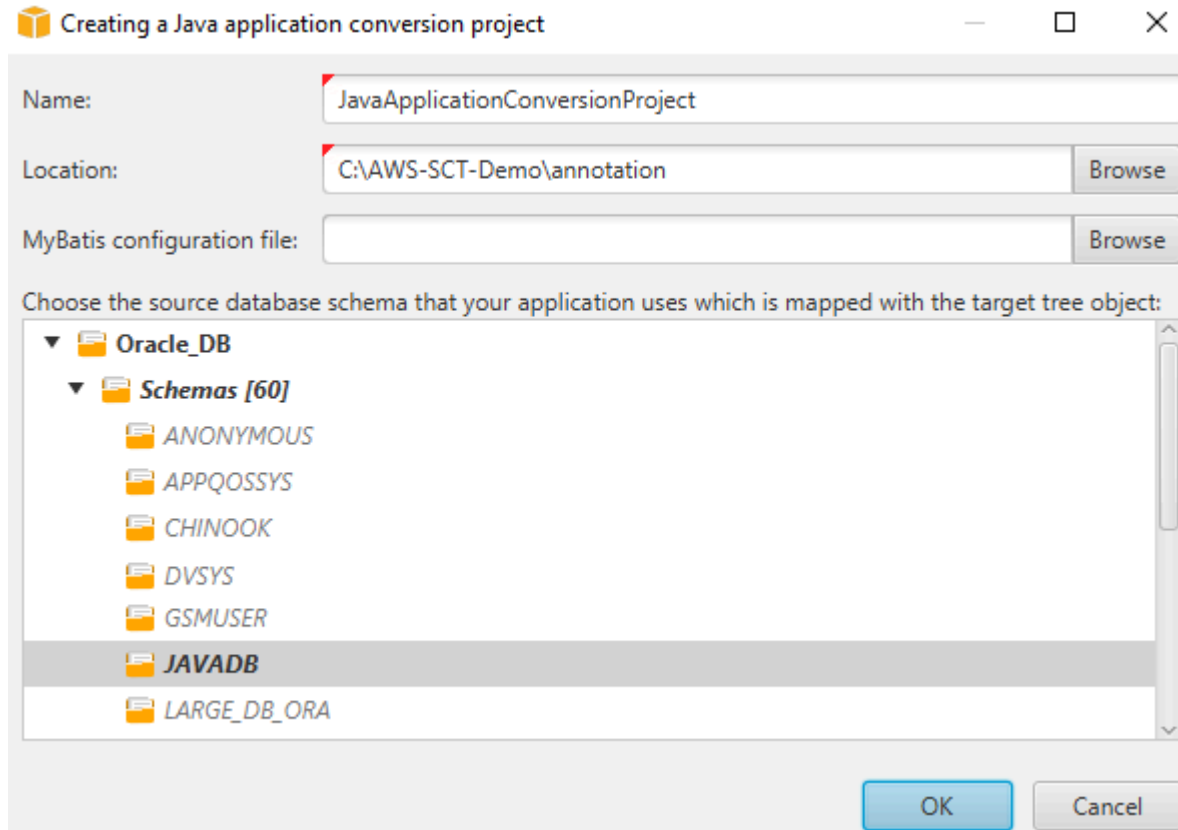
Sie können mehrere Projekte zur Anwendungskonvertierung zu einem einzigen hinzufügen AWS SCT Projekt. Gehen Sie wie folgt vor, um ein Konvertierungsprojekt für Java-Anwendungen zu erstellen.

Um ein Projekt zur Konvertierung von Java-Anwendungen zu erstellen

1. Erstellen Sie ein Datenbankkonvertierungsprojekt und fügen Sie eine Oracle-Quelldatenbank hinzu. Weitere Informationen erhalten Sie unter [Ein AWS SCT Projekt erstellen](#) und [Hinzufügen von Datenbankservern zu einem AWS SCT Projekt](#).
2. Fügen Sie eine Zuordnungsregel hinzu, die Ihre Oracle-Quelldatenbank und eine PostgreSQL-Zieldatenbank umfasst. Sie können eine PostgreSQL-Zieldatenbank hinzufügen oder eine virtuelle PostgreSQL-Zieldatenbankplattform in einer Zuordnungsregel verwenden. Weitere Informationen erhalten Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#) und [Verwendung virtueller Ziele](#).

3. Auf der AnsichtMenü, wählen Hauptgröße.
4. Auf der BewerbungenMenü, wählen Neue Java-Anwendung.

Das Erstellen eines Projekts zur Konvertierung von Java-Anwendungen in ein Dialogfeld wird angezeigt.



5. Für Name, geben Sie einen Namen für Ihr Java-Anwendungskonvertierungsprojekt ein. Da jedes Datenbankschema-Konvertierungsprojekt ein oder mehrere untergeordnete Anwendungskonvertierungsprojekte haben kann, wählen Sie einen Namen, der Sinn macht, wenn Sie mehrere Projekte hinzufügen.
6. Für Ort, geben Sie die Position des Quellcodes Ihrer Anwendung ein.
7. (Optional) Für MyBatis Konfigurationsdatei, geben Sie den Pfad zur MyBatis Konfigurationsdatei. AWS SCT scannt Ihren Anwendungsordner, um diese Datei automatisch zu finden. Wenn sich diese Datei nicht in Ihrem Anwendungsordner befindet oder wenn Sie mehrere Konfigurationsdateien verwenden, geben Sie den Pfad manuell ein.
8. Wählen Sie im Quellbaum das Schema aus, das Ihre Anwendung verwendet. Stellen Sie sicher, dass dieses Schema Teil einer Zuordnungsregel ist. AWS SCT hebt die Schemas, die Teil einer Zuordnungsregel sind, fett hervor.
9. Wählen OK um Ihr Projekt zur Konvertierung von Java-Anwendungen zu erstellen.

10. Finden Sie Ihr Projekt zur Konvertierung von Java-Anwendungen imAnwendungenKnoten im linken Bereich.

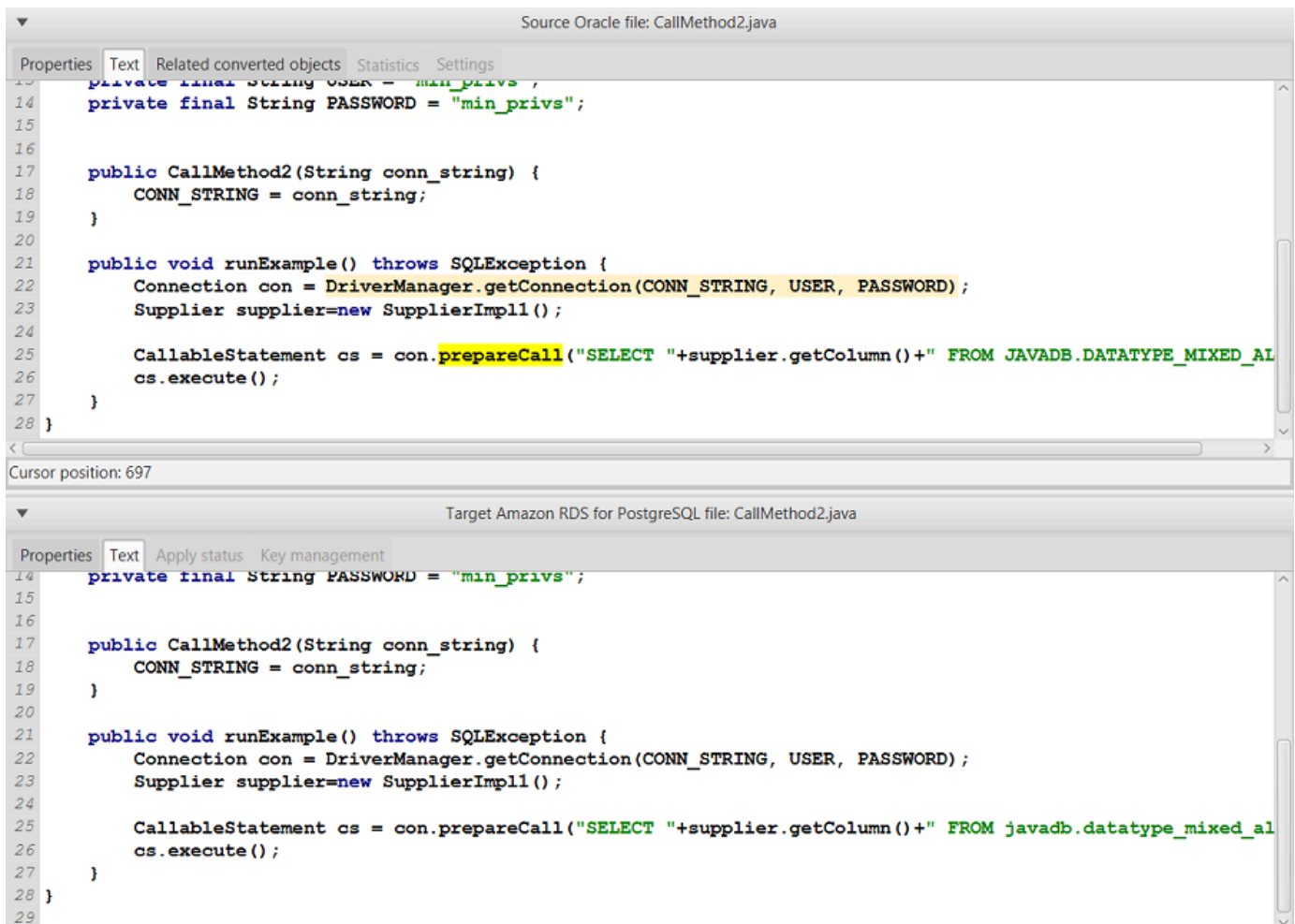
Konvertieren Sie den SQL-Code Ihrer Java-Anwendung inAWS SCT

Nachdem Sie Ihre Java-Anwendung zur Datei hinzugefügt habenAWS SCTProjekt, konvertieren Sie den SQL-Code aus dieser Anwendung in ein Format, das mit Ihrer Zieldatenbankplattform kompatibel ist. Verwenden Sie das folgende Verfahren, um den in Ihre Java-Anwendung eingebetteten SQL-Code in derAWS Schema Conversion Tool.

Um Ihren SQL-Code zu konvertieren

1. Erweitern Sie dasJavaKnoten unterAnwendungenim linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie konvertieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. WählenKonvertiere.AWS SCTanalysiert Ihre Quellcodedateien, bestimmt die Anwendungslogik und lädt Code-Metadaten in das Projekt. Diese Code-Metadaten umfassen Java-Klassen, Objekte, Methoden, globale Variablen, Schnittstellen usw.

Im Zieldatenbank-PanelAWS SCTerstellt eine ähnliche Ordnerstruktur wie Ihr Quellanwendungsprojekt. Hier können Sie den konvertierten Anwendungscode überprüfen.



```
Source Oracle file: CallMethod2.java
13 private final String USER = "min_privs";
14 private final String PASSWORD = "min_privs";
15
16
17 public CallMethod2(String conn_string) {
18     CONN_STRING = conn_string;
19 }
20
21 public void runExample() throws SQLException {
22     Connection con = DriverManager.getConnection(CONN_STRING, USER, PASSWORD);
23     Supplier supplier=new SupplierImpl();
24
25     CallableStatement cs = con.prepareCall("SELECT "+supplier.getColumn()+" FROM JAVADB.DATATYPE_MIXED_AL
26     cs.execute();
27 }
28 }
Cursor position: 697

Target Amazon RDS for PostgreSQL file: CallMethod2.java
14 private final String PASSWORD = "min_privs";
15
16
17 public CallMethod2(String conn_string) {
18     CONN_STRING = conn_string;
19 }
20
21 public void runExample() throws SQLException {
22     Connection con = DriverManager.getConnection(CONN_STRING, USER, PASSWORD);
23     Supplier supplier=new SupplierImpl();
24
25     CallableStatement cs = con.prepareCall("SELECT "+supplier.getColumn()+" FROM javadb.datatype_mixed_al
26     cs.execute();
27 }
28 }
29 }
```

4. Speichern Sie Ihren konvertierten Anwendungscode. Weitere Informationen finden Sie unter [Speichern Sie Ihren konvertierten Anwendungscode](#).

Ihre Java-Anwendungen können SQL-Code enthalten, der mit verschiedenen Quelldatenbanken interagiert. Sie können mehrere dieser Quelldatenbanken zu PostgreSQL migrieren. Stellen Sie in diesem Fall sicher, dass Sie keinen SQL-Code konvertieren, der mit Datenbanken interagiert, die Sie aus dem Migrationsbereich ausgeschlossen haben. Sie können Quelldateien Ihrer Java-Anwendung vom Konvertierungsbereich ausschließen. Deaktivieren Sie dazu die Kontrollkästchen für die Namen der Dateien, die Sie aus dem Konvertierungsbereich ausschließen möchten.

Nachdem Sie den Konvertierungsbereich geändert haben, AWS SCT analysiert weiterhin den SQL-Code aller Quelldateien Ihrer Java-Anwendungen. Dann AWS SCT kopiert alle Quelldateien, die Sie aus dem Konvertierungsbereich ausgeschlossen haben, in den Zielordner. Dieser Vorgang ermöglicht es, Ihre Anwendung zu erstellen, nachdem Sie Ihre konvertierten Anwendungsdateien gespeichert haben.

Speichern Sie Ihren konvertierten Anwendungscode mit AWS SCT

Mit dem folgenden Verfahren können Sie Ihren konvertierten Anwendungscode speichern.

Um Ihren konvertierten Anwendungscode zu speichern

1. Erweitern Sie das Java-Knoten unter Anwendungen im Zieldatenbank-Panel.
2. Wählen Sie Ihre konvertierte Anwendung und wählen Sie Speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem der konvertierte Anwendungscode gespeichert werden soll, und wählen Sie Wählen Sie einen Ordner.

Wenn Ihre Java-Quellanwendung die verwendet MyBatis Framework, stellen Sie sicher, dass Sie Ihre Konfigurationsdatei aktualisieren, damit sie mit Ihrer neuen Datenbank funktioniert.

Verwaltung von Projekten zur Konvertierung von Java-Anwendungen in AWS SCT

Sie können mehrere Projekte zur Konvertierung von Java-Anwendungen hinzufügen und den Anwendungscode in der Datei aktualisieren AWS SCT Projekt oder entfernen Sie ein Java-Konvertierungsprojekt aus Ihrem AWS SCT Projekt

Um ein zusätzliches Projekt zur Konvertierung von Java-Anwendungen hinzuzufügen

1. Erweitern Sie das Anwendungen-Knoten im linken Bereich.
2. Wählen Sie den Java-Node, und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie New application (Neue Anwendung) aus.
4. Geben Sie die Informationen ein, die zum Erstellen eines neuen Projekts zur Konvertierung von Java-Anwendungen erforderlich sind. Weitere Informationen finden Sie unter [Projekte zur Konvertierung von Java-Anwendungen erstellen](#).

Nachdem Sie Änderungen an Ihrem Quellanwendungscode vorgenommen haben, laden Sie ihn in den AWS SCT Projekt

Um den aktualisierten Anwendungscode hochzuladen

1. Erweitern Sie das Java-Knoten unter Anwendungen im linken Bereich.

2. Wählen Sie die Anwendung aus, die Sie aktualisieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen Auffrischen und dann wähle Ja.

AWS SCT lädt Ihren Anwendungscode aus den Quelldateien hoch und entfernt die Konvertierungsergebnisse. Um Codeänderungen beizubehalten, die Sie in vorgenommen haben, erstellen Sie zusammen mit den Konvertierungsergebnissen ein neues Java-Konvertierungsprojekt.

Wenn Ihre Java-Quellanwendung das verwendet MyBatis Rahmen, verwendet die MyBatis Konfigurationsdatei, um Ihren SQL-Code zu analysieren. Nachdem Sie diese Datei geändert haben, laden Sie sie in die AWS SCT Projekt

Um den Pfad zu der MyBatis Konfigurationsdatei

1. Erweitern Sie die Java Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie Ihre Anwendung und wählen Sie dann Einstellungen.
3. Wählen Stöbern, und wählen Sie dann MyBatis Konfigurationsdatei.
4. Wählen Sie Apply (Anwenden) aus.
5. Wählen Sie im linken Bereich aus, öffnen Sie das Kontextmenü (rechte Maustaste) und wählen Sie Aktualisieren.

Um ein Projekt zur Konvertierung einer Java-Anwendung zu entfernen

1. Erweitern Sie das Java Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie entfernen möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wähle Löschen und dann wähle OKAY.

Erstellen Sie einen Bewertungsbericht zur Konvertierung von Java-Anwendungen in AWS SCT

Der Bewertungsbericht zur Konvertierung von Java-Anwendungen enthält Informationen zur Konvertierung des in Ihrer Java-Anwendung eingebetteten SQL-Codes in ein Format, das mit Ihrer Zieldatenbank kompatibel ist. Der Bewertungsbericht enthält Konvertierungsdetails für alle SQL-

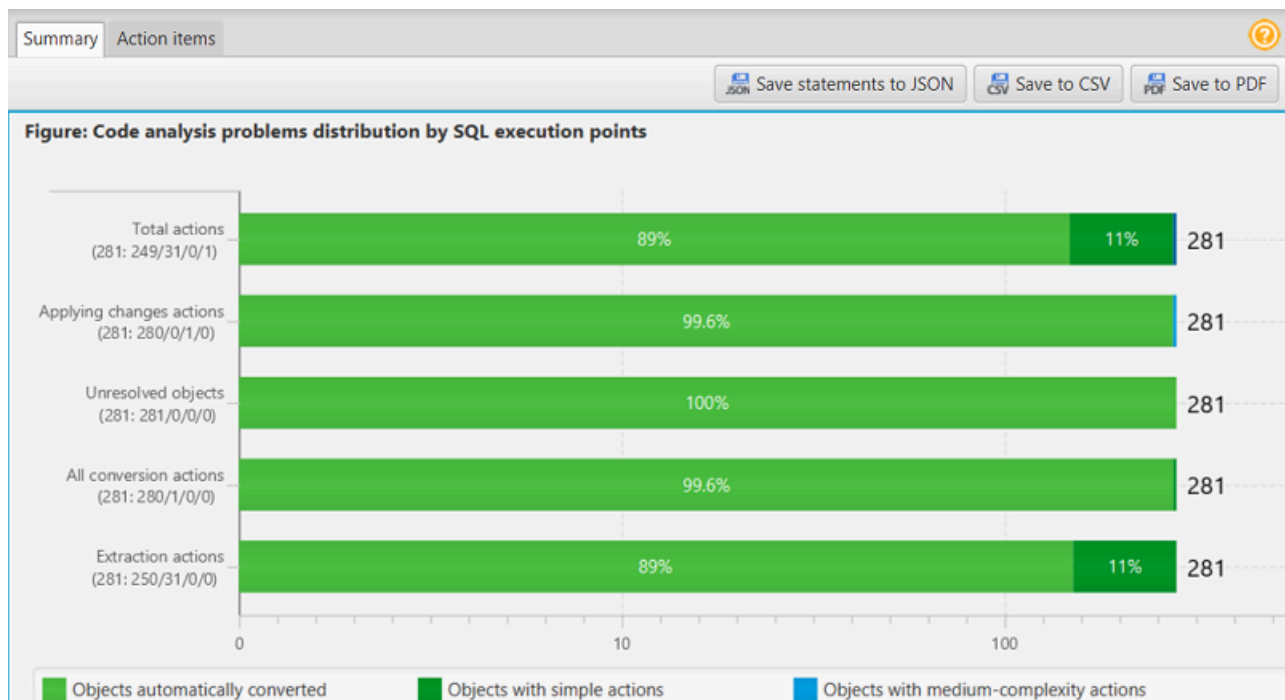
Ausführungspunkte und alle Quellcodedateien. Der Bewertungsbericht enthält auch Aktionspunkte für SQL-Code, die AWS SCT nicht konvertieren.

Gehen Sie wie folgt vor, um einen Bewertungsbericht zur Konvertierung von Java-Anwendungen zu erstellen.

So erstellen Sie einen Bewertungsbericht zur Konvertierung von Java-Anwendungen

1. Erweitern Sie JavaKnoten unter Anwendungen im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie konvertieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wähle Konvertiere.
4. Auf der Ansicht Menü, wählen Ansicht des Bewertungsberichts.
5. Prüfen Sie die Registerkarte Summary.

Das Zusammenfassung Auf der Registerkarte (unten) werden die zusammenfassenden Informationen aus dem Bewertungsbericht für Java-Anwendungen angezeigt. Es zeigt die Konvertierungsergebnisse für alle SQL-Ausführungspunkte und alle Quellcodedateien.



6. Wählen Sie Speichern Sie Anweisungen in JSON um den extrahierten SQL-Code aus Ihrer Java-Anwendung als JSON-Datei zu speichern.
7. (Optional) Speichern Sie eine lokale Kopie des Berichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):

- Wählen **Speichern** oben rechts, um den Bericht als PDF-Datei zu speichern.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Anwendungskonvertierung.

- Wählen Sie **Speichern der Ordnergröße** oben rechts, um den Bericht als CSV-Datei zu speichern.

Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung des SQL-Codes erforderlich ist.

Konvertierung von SQL-Code in Pro*C-Anwendungen mit AWS SCT

Für eine Konvertierung von Oracle nach PostgreSQL können Sie den AWS Schema Conversion Tool (AWS SCT) um in Ihre Pro*C-Anwendungen eingebetteten SQL-Code zu konvertieren. Dieser spezielle Pro*C-Anwendungskonverter versteht die Anwendungslogik. Er sammelt Anweisungen, die sich in verschiedenen Anwendungsobjekten befinden, wie Funktionen, Parameter, lokale Variablen usw.

Aufgrund dieser gründlichen Analyse liefert der SQL-Code-Konverter für Pro*C-Anwendungen bessere Konvertierungsergebnisse als der generische Konverter.

Projekte zur Konvertierung von Pro*C-Anwendungen erstellen in AWS SCT

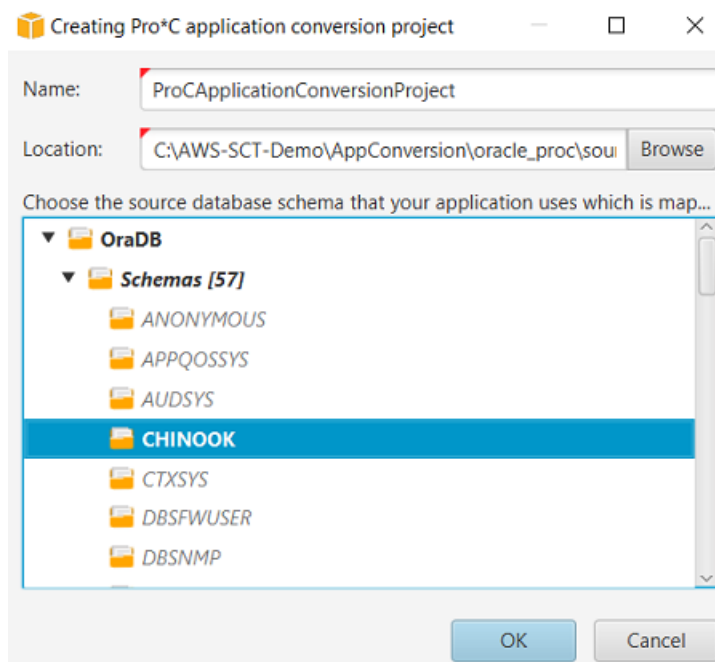
Sie können ein Pro*C-Anwendungskonvertierungsprojekt nur für die Konvertierung von Oracle-Datenbankschemas in PostgreSQL-Datenbankschemas erstellen. Stellen Sie sicher, dass Sie Ihrem Projekt eine Zuordnungsregel hinzufügen, die ein Oracle-Quellschema und eine PostgreSQL-Zieldatenbank enthält. Weitere Informationen finden Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#).

Sie können mehrere Projekte zur Anwendungskonvertierung zu einem einzigen hinzufügen AWS SCT Projekt. Gehen Sie wie folgt vor, um ein Projekt zur Konvertierung von Pro*C-Anwendungen zu erstellen.

Um ein Projekt zur Konvertierung von Pro*C-Anwendungen zu erstellen

1. Erstellen Sie ein Datenbankkonvertierungsprojekt und fügen Sie eine Oracle-Quelldatenbank hinzu. Weitere Informationen erhalten Sie unter [Ein AWS SCT Projekt erstellen](#) und [Hinzufügen von Datenbankservern zu einem AWS SCT Projekt](#).
2. Fügen Sie eine Zuordnungsregel hinzu, die Ihre Oracle-Quelldatenbank und eine PostgreSQL-Zieldatenbank umfasst. Sie können eine PostgreSQL-Zieldatenbank hinzufügen oder eine virtuelle PostgreSQL-Zieldatenbankplattform in einer Zuordnungsregel verwenden. Weitere Informationen erhalten Sie unter [Erstellen von Zuordnungsregeln in AWS SCT](#) und [Verwendung virtueller Ziele](#).
3. Auf der Ansicht Menü, wählen Hauptgröße.
4. Auf der Bewerbungen Menü, wählen Neue Pro*C-Anwendung.

Das Ein Projekt zur Konvertierung von Pro*C-Anwendungen erstellen ein Dialogfeld wird angezeigt.



5. Für Name, geben Sie einen Namen für Ihr Pro*C-Anwendungskonvertierungsprojekt ein. Da jedes Datenbankschema-Konvertierungsprojekt ein oder mehrere untergeordnete Anwendungskonvertierungsprojekte haben kann, wählen Sie einen Namen, der Sinn macht, wenn Sie mehrere Projekte hinzufügen.
6. Für Ort, geben Sie die Position des Quellcodes Ihrer Anwendung ein.

7. Wählen Sie im Quellbaum das Schema aus, das Ihre Anwendung verwendet. Stellen Sie sicher, dass dieses Schema Teil einer Zuordnungsregel ist. AWS SCT hebt die Schemas, die Teil einer Zuordnungsregel sind, fett hervor.
8. Wählen OKAY um Ihr Projekt zur Konvertierung Ihrer Pro*C-Anwendung zu erstellen.
9. Finden Sie Ihr Projekt zur Konvertierung Ihrer Pro*C-Anwendung im AnwendungenKnoten im linken Bereich.

Konvertieren Sie den SQL-Code Ihrer Pro*C-Anwendung in AWS SCT

Nachdem Sie Ihre Pro*C-Anwendung zur AWS SCT Projekt, konvertieren Sie den SQL-Code aus dieser Anwendung in ein Format, das mit Ihrer Zieldatenbankplattform kompatibel ist. Verwenden Sie das folgende Verfahren, um den in Ihre Pro*C-Anwendung eingebetteten SQL-Code in der AWS Schema Conversion Tool.

Um Ihren SQL-Code zu konvertieren

1. Erweitern Sie das Pro*C Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die zu konvertierende Anwendung aus und wählen Sie dann Einstellungen.
 - a. Für Globaler Header-Dateipfad, geben Sie den Pfad zu den Header-Dateien ein, die Ihr Anwendungsprojekt verwendet.
 - b. Wählen Interpretiere alle ungelösten Hostvariablen als um alle ungelösten Variablen im konvertierten Code zu sehen.
 - c. Wählen Verwenden Sie die Konvertierungsfunktion für Zeichenketten mit fester Breite aus dem Erweiterungspaket um die Funktionen des Erweiterungspakets im konvertierten SQL-Code zu verwenden. AWS SCT schließt die Erweiterungspaketdateien in Ihr Anwendungsprojekt ein.
 - d. Wählen Transformieren Sie anonyme PL/SQL-Blöcke in eigenständige SQL-Aufrufe oder gespeicherte Funktionen um gespeicherte Prozeduren in Ihrer Zieldatenbank für alle anonymen PL/SQL-Blöcke zu erstellen. AWS SCT schließt dann die Ausführungen dieser gespeicherten Prozeduren in den konvertierten Anwendungscode ein.
 - e. Wählen Sie Benutzerdefinierten Cursorfluss verwenden um die Konvertierung von Oracle-Datenbank cursorn zu verbessern.
3. Wählen Sie im linken Bereich die Anwendung aus, die Sie konvertieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).

- Wählen Konvertiere. AWS SCT analysiert Ihre Quellcode-Dateien, bestimmt die Anwendungslogik und lädt Code-Metadaten in das Projekt. Diese Code-Metadaten umfassen Pro*C-Klassen, Objekte, Methoden, globale Variablen, Schnittstellen usw.

Im Zieldatenbank-Panel AWS SCT erstellt eine ähnliche Ordnerstruktur wie Ihr Quellanwendungsprojekt. Hier können Sie den konvertierten Anwendungscode überprüfen.

The screenshot displays two panels side-by-side. The top panel, titled 'Source Oracle app function: main() int', shows the original Pro*C code with line numbers 11 through 20. The code includes a variable declaration, a comment, and SQL execution statements. The bottom panel, titled 'Target Amazon RDS for PostgreSQL app function: main() int', shows the converted SQL code with line numbers 8 through 18. The converted code uses 'EXEC SQL' statements to execute the same logic in a PostgreSQL environment.

```

Source Oracle app function: main() int
Properties Text Related converted objects Statistics Settings
11  int i = 5;
12
13  /* Connect string */
14
15  EXEC SQL INSERT INTO embeddedc.t_insert(i) VALUES (:i);
16
17  EXEC SQL COMMIT;
18
19  return 0;
20 }
Cursor position: 118 Click position: 197

Target Amazon RDS for PostgreSQL app function: main() int
Properties Text Apply status Key management
8
9  EXEC SQL int i = 5;
10
11  /* Connect string */
12
13  EXEC SQL INSERT INTO embeddedc.t_insert (i)
14  VALUES (:i);
15
16  EXEC SQL COMMIT;
17
18  return 0;

```

- Speichern Sie Ihren konvertierten Anwendungscode. Weitere Informationen finden Sie unter [Bearbeiten und Speichern Ihres konvertierten Anwendungscodes](#).

Bearbeiten und speichern Sie Ihren konvertierten Anwendungscode mit AWS SCT

Sie können die konvertierten SQL-Anweisungen bearbeiten und verwenden AWS SCT um diesen bearbeiteten Code in den konvertierten Pro*C-Anwendungscode einzubetten. Mit dem folgenden Verfahren können Sie Ihren konvertierten SQL-Code bearbeiten.

Um Ihren konvertierten SQL-Code zu bearbeiten

- Erweitern Sie das Pro*C-Knoten unter Anwendungen im linken Bereich.
- Wählen Sie die Anwendung aus, die Sie konvertieren möchten, öffnen Sie das Kontextmenü (rechte Maustaste) und wählen Sie Konvertieren.

3. Auf der AnsichtMenü, wählen Ansicht des Bewertungsberichts.
4. Wähle Kontoauszüge im CSV-Format speichern um den extrahierten SQL-Code aus Ihrer Pro*C-Anwendung als CSV-Datei zu speichern.
5. Geben Sie den Namen der CSV-Datei ein, um den extrahierten SQL-Code zu speichern, und wählen Sie Speichern.
6. Bearbeiten Sie den extrahierten SQL-Code.
7. Auf dem AnsichtMenü, wählen Hauptgröße.
8. Erweitern Sie das Pro*CKnoten unter Anwendungen im Zieldatenbank-Panel.
9. Wählen Sie die konvertierte Anwendung aus, öffnen Sie das Kontextmenü (rechte Maustaste) und wählen Sie Kontoauszüge aus CSV importieren.
10. Wähle Ja, wählen Sie dann die Datei mit Ihrem bearbeiteten SQL-Code und wählen Öffnen.

AWS SCT zerlegt die konvertierten SQL-Anweisungen in Teile und platziert sie in den entsprechenden Objekten Ihres Quellanwendungscode. Mit dem folgenden Verfahren können Sie Ihren konvertierten Anwendungscode speichern.

Um Ihren konvertierten Anwendungscode zu speichern

1. Erweitern Sie das Pro*CKnoten unter Anwendungen im Zieldatenbank-Panel.
2. Wählen Sie Ihre konvertierte Anwendung und wählen Sie Speichern.
3. Geben Sie den Pfad zu dem Ordner ein, in dem der konvertierte Anwendungscode gespeichert werden soll, und wählen Sie Wählen Sie einen Ordner.

Verwaltung von Projekten zur Konvertierung von Pro*C-Anwendungen in AWS SCT

Sie können mehrere Projekte zur Konvertierung von Pro*C-Anwendungen hinzufügen und den Anwendungscode in der Datei aktualisieren AWS SCT Projekt oder entfernen Sie ein Pro*C-Konvertierungsprojekt aus Ihrem AWS SCT projekt

Um ein zusätzliches Projekt zur Konvertierung von Pro*C-Anwendungen hinzuzufügen

1. Erweitern Sie das AnwendungenKnoten im linken Bereich.
2. Wählen Sie den Pro*CNode, und öffnen Sie das Kontextmenü (Rechtsklick).
3. Wählen Sie New application (Neue Anwendung) aus.

4. Geben Sie die Informationen ein, die für die Erstellung eines neuen Projekts zur Konvertierung von Pro*C-Anwendungen erforderlich sind. Weitere Informationen finden Sie unter [Projekte zur Konvertierung von Pro*C-Anwendungen erstellen](#).

Nachdem Sie Änderungen an Ihrem Quellanwendungscode vorgenommen haben, laden Sie ihn in den AWS SCT-Projekt

Um den aktualisierten Anwendungscode hochzuladen

1. Erweitern Sie Pro*C-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie aktualisieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen Auffrischen und dann wähle Ja.

AWS SCT lädt Ihren Anwendungscode aus den Quelldateien hoch und entfernt die Konvertierungsergebnisse. Um Codeänderungen beizubehalten, die Sie in vorgenommen haben, AWS SCT und die Konvertierungsergebnisse, erstellen Sie ein neues Pro*C-Konvertierungsprojekt.

Um ein Projekt zur Konvertierung einer Pro*C-Anwendung zu entfernen

1. Erweitern Sie das Pro*C-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie entfernen möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen Löschen und dann wähle OKAY.

Erstellen eines Bewertungsberichts zur Konvertierung von Pro*C-Anwendungen in AWS SCT

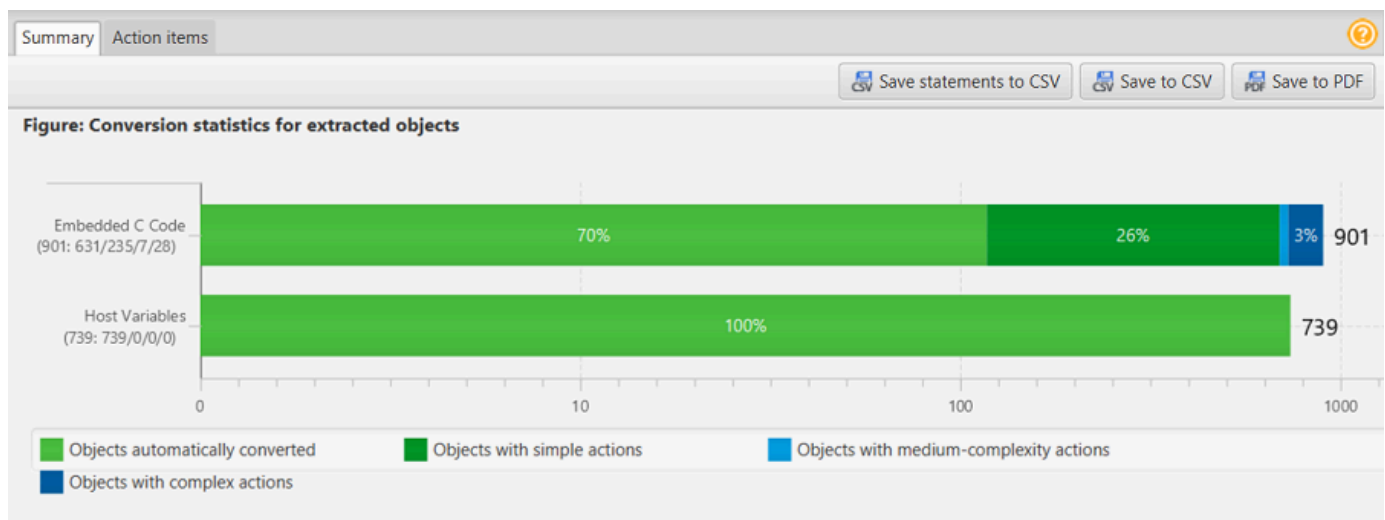
Der Bewertungsbericht zur Konvertierung von Pro*C-Anwendungen enthält Informationen zur Konvertierung des in Ihrer Pro*C-Anwendung eingebetteten SQL-Codes in ein Format, das mit Ihrer Zieldatenbank kompatibel ist. Der Bewertungsbericht enthält Konvertierungsdetails für alle SQL-Ausführungspunkte und alle Quellcodedateien. Der Bewertungsbericht enthält auch Aktionspunkte für SQL-Code, die AWS SCT kann nicht konvertieren.

Gehen Sie wie folgt vor, um einen Bewertungsbericht zur Konvertierung von Pro*C-Anwendungen zu erstellen.

So erstellen Sie einen Bewertungsbericht zur Konvertierung von Pro*C-Anwendungen

1. Erweitern Sie Pro*C-Knoten unter Anwendungen im linken Bereich.
2. Wählen Sie die Anwendung aus, die Sie konvertieren möchten, und öffnen Sie das Kontextmenü (rechte Maustaste).
3. Wählen Konvertiere.
4. Auf der Ansicht Menü, wählen Ansicht des Bewertungsberichts.
5. Prüfen Sie die Registerkarte Summary.

Das Zusammenfassung Auf der Registerkarte (unten) werden die zusammenfassenden Informationen aus dem Bewertungsbericht für die Pro*C-Anwendung angezeigt. Es zeigt die Konvertierungsergebnisse für alle SQL-Ausführungspunkte und alle Quellcodedateien.



6. Wählen Sie Kontoauszüge im CSV-Format speichern um den extrahierten SQL-Code aus der Pro*C-Anwendung als Datei mit kommagetrennten Werten (CSV) zu speichern.
7. (Optional) Speichern Sie eine lokale Kopie des Berichts entweder als PDF-Datei oder als Datei mit kommagetrennten Werten (CSV):
 - Wählen Speichern oben rechts, um den Bericht als PDF-Datei zu speichern.

Die PDF-Datei enthält die Zusammenfassung, Aktionspunkte und Empfehlungen für die Anwendungskonvertierung.

- Wählen Sie Speichern der Ordnergröße oben rechts, um den Bericht als CSV-Datei zu speichern.

Die CSV-Datei enthält Aktionspunkte, empfohlene Maßnahmen und eine geschätzte Komplexität des manuellen Aufwands, der für die Konvertierung des SQL-Codes erforderlich ist.

Verwenden von AWS SCT Erweiterungspaketen

Ein AWS SCT Erweiterungspaket ist ein Zusatzmodul, das Funktionen emuliert, die in einer Quelldatenbank vorhanden sind und für die Konvertierung von Objekten in die Zieldatenbank erforderlich sind. Bevor Sie ein AWS SCT Erweiterungspaket installieren können, konvertieren Sie Ihr Datenbankschema.

Jedes AWS SCT Erweiterungspaket umfasst die folgenden Komponenten:

- **DB-Schema** — Beinhaltet SQL-Funktionen, -Prozeduren und -Tabellen zur Emulation bestimmter Datenbankobjekte wie Sequenzen im Bereich Online-Transaktionsverarbeitung (OLTP) und Online-Analytische Verarbeitung (OLAP). Emuliert außerdem nicht von der Quelldatenbank aus. built-in-functions Der Name dieses Schemas hat das folgende Format: `aws_<database_engine_name>_ext`
- **AWS Lambda Funktionen (für bestimmte OLTP-Datenbanken)** — Beinhaltet AWS Lambda Funktionen, die komplexe Datenbankfunktionen emulieren, z. B. die Auftragsplanung und das Senden von E-Mails.
- **Benutzerdefinierte Bibliotheken für OLAP-Datenbanken** — Enthält eine Reihe von Java- und Python-Bibliotheken, mit denen Sie Microsoft SQL Server Integration Services (SSIS) -Skripts zum Extrahieren, Transformieren und Laden (ETL) nach AWS Glue oder migrieren können. AWS Glue Studio

Java-Bibliotheken umfassen die folgenden Module:

- `spark-excel_2.11-0.13.1.jar` — Um die Funktionalität von Excel-Quell- und Zielkomponenten zu emulieren.
- `spark-xml_2.11-0.9.0.jar`, `poi-ooxml-schemas-4.1.2.jar`, und `xmlbeans-3.1.0.jar` — Um die Funktionalität der XML-Quellkomponente zu emulieren.

Python-Bibliotheken umfassen die folgenden Module:

- `sct_utils.py` — Um Quelldatentypen zu emulieren und Parameter für die Spark-SQL-Abfrage vorzubereiten.
- `ssis_datetime.py` — Um integrierte Funktionen für Datum und Uhrzeit zu emulieren.
- `ssis_null.py` — Um die ISNULL und die REPLACENULL integrierten Funktionen zu emulieren.
- `ssis_string.py` — Um eingebaute Zeichenkettenfunktionen zu emulieren.

Weitere Informationen zu diesen Bibliotheken finden Sie unter [Verwenden von benutzerdefinierten Bibliotheken für AWS SCT Erweiterungspakete](#).

Sie können AWS SCT Erweiterungspakete auf zwei Arten anwenden:

- AWS SCT kann ein Erweiterungspaket automatisch anwenden, wenn Sie ein Zieldatenbankskript anwenden, indem Sie im Kontextmenü die Option Auf Datenbank anwenden wählen. AWS SCT wendet das Erweiterungspaket an, bevor es alle anderen Schemaobjekte anwendet.
- Um ein Erweiterungspaket manuell anzuwenden, wählen Sie die Zieldatenbank aus und wählen Sie dann im Kontextmenü (Rechtsklick) die Option Erweiterungspaket anwenden für aus. In den meisten Situationen ist eine automatische Anwendung ausreichend. Möglicherweise möchten Sie das Paket jedoch manuell anwenden, falls es versehentlich gelöscht wurde.

Jedes Mal, wenn Sie ein AWS SCT Erweiterungspaket auf einen Zieldatenspeicher anwenden, werden die Komponenten überschrieben, und es AWS SCT wird eine entsprechende Benachrichtigung angezeigt. Um diese Benachrichtigungen zu deaktivieren, wählen Sie Einstellungen, Allgemeine Einstellungen, Benachrichtigungen und dann die Option Warnung zum Austausch des Erweiterungspakets ausblenden aus.

Für eine Konvertierung von Microsoft SQL Server zu PostgreSQL können Sie das SQL Server to PostgreSQL Extension Pack in verwenden. AWS SCT Dieses Erweiterungspaket emuliert SQL Server Agent und SQL Server Database Mail. Weitere Informationen finden Sie unter [Emulieren des SQL Server-Agents in PostgreSQL mit einem Erweiterungspaket](#) und [Emulieren von SQL Server-Datenbank-Mail in PostgreSQL mit einem Erweiterungspaket](#).

Im Folgenden finden Sie weitere Informationen zum Arbeiten mit AWS SCT Erweiterungspaketen.

Themen

- [Berechtigungen für die Verwendung des AWS SCT Erweiterungspakets](#)
- [Verwenden Sie das Erweiterungspaket-Schema](#)
- [Verwenden von benutzerdefinierten Bibliotheken für AWS SCT Erweiterungspakete](#)
- [Verwenden Sie die AWS Lambda Funktionen aus dem AWS SCT Erweiterungspaket](#)
- [Funktionen für das AWS SCT Erweiterungspaket konfigurieren](#)

Berechtigungen für die Verwendung des AWS SCT Erweiterungspakets

Das AWS SCT Erweiterungspaket für Amazon Aurora emuliert das Senden von E-Mails, die Auftragsplanung, das Einstellen von Warteschlangen und andere Operationen mithilfe von AWS Lambda Funktionen. Wenn Sie das AWS SCT Erweiterungspaket auf Ihre Aurora-Zieldatenbank anwenden, AWS SCT erstellt es eine neue AWS Identity and Access Management (IAM-) Rolle und eine Inline-IAM-Richtlinie. Als Nächstes AWS SCT erstellt es eine neue Lambda-Funktion und konfiguriert Ihren Aurora-DB-Cluster für ausgehende Verbindungen zu AWS Lambda. Um diese Operationen auszuführen, stellen Sie sicher, dass Sie Ihrem IAM-Benutzer die folgenden erforderlichen Berechtigungen gewähren:

- `iam:CreateRole`— um eine neue IAM-Rolle für Ihr AWS Konto zu erstellen.
- `iam:CreatePolicy`— um eine neue IAM-Richtlinie für Ihr AWS Konto zu erstellen.
- `iam:AttachRolePolicy`— um die angegebene Richtlinie an Ihre IAM-Rolle anzuhängen.
- `iam:PutRolePolicy`— um ein integriertes Richtliniendokument zu aktualisieren, das in Ihre IAM-Rolle eingebettet ist.
- `iam:PassRole`— um die angegebene IAM-Rolle an die Regel-Engine zu übergeben.
- `iam:TagRole`— um einer IAM-Rolle Tags hinzuzufügen.
- `iam:TagPolicy`— um einer IAM-Richtlinie Tags hinzuzufügen.
- `lambda:ListFunctions`— um die Liste Ihrer Lambda-Funktionen zu sehen.
- `lambda:ListTags`— um die Liste der Tags Ihrer Lambda-Funktionen zu sehen.
- `lambda:CreateFunction`— um eine neue Lambda-Funktion zu erstellen.
- `rds:AddRoleToDBCluster`— um Ihrem Aurora-DB-Cluster eine IAM-Rolle zuzuordnen.

Das AWS SCT Erweiterungspaket für Amazon Redshift emuliert Quell-Data Warehouse-Basisfunktionen, die erforderlich sind, wenn konvertierte Objekte auf Amazon Redshift angewendet werden. Bevor Sie Ihren konvertierten Code auf Amazon Redshift anwenden, müssen Sie das Erweiterungspaket für Amazon Redshift anwenden. Nehmen Sie dazu die `iam:SimulatePrincipalPolicy` Aktion in Ihre IAM-Richtlinie auf.

AWS SCT verwendet den IAM Policy Simulator, um die erforderlichen Berechtigungen für die Installation des Amazon Redshift Redshift-Erweiterungspakets zu überprüfen. Der IAM Policy Simulator kann eine Fehlermeldung anzeigen, auch wenn Sie Ihren IAM-Benutzer korrekt konfiguriert

haben. Dies ist ein bekanntes Problem des IAM Policy Simulator. Außerdem zeigt der IAM-Richtliniensimulator eine Fehlermeldung an, wenn Sie die `iam:SimulatePrincipalPolicy` Aktion nicht in Ihrer IAM-Richtlinie haben. In diesen Fällen können Sie die Fehlermeldung ignorieren und das Erweiterungspaket mithilfe des Erweiterungspaket-Assistenten anwenden. Weitere Informationen finden Sie unter [Anwenden des Erweiterungspakets](#).

Verwenden Sie das Erweiterungspaket-Schema

Wenn Sie Ihr Datenbank- oder Data Warehouse-Schema konvertieren, fügt AWS SCT ein zusätzliches Schema zur Zieldatenbank hinzu. Über dieses Schema werden SQL-Systemfunktionen der Quelldatenbank implementiert, die zum Schreiben des konvertierten Schemas in die Zieldatenbank benötigt werden. Dieses Zusätzliche Schema wird als Erweiterungspaketschema bezeichnet.

Das Erweiterungspaketschema für OLTP-Datenbanken wird wie folgt entsprechend der Quelldatenbank benannt:

- Microsoft SQL Server: `AWS_SQLSERVER_EXT`
- MySQL: `AWS_MYSQL_EXT`
- Oracle: `AWS_ORACLE_EXT`
- PostgreSQL: `AWS_POSTGRESQL_EXT`

Das Erweiterungspaketschema für OLAP-Datawarehouse-Anwendungen wird wie folgt entsprechend dem Quelldatenspeicher benannt:

- Greenplum: `AWS_GREENPLUM_EXT`
- Microsoft SQL Server: `AWS_SQLSERVER_EXT`
- Netezza: `AWS_NETEZZA_EXT`
- Oracle: `AWS_ORACLE_EXT`
- Teradata: `AWS_TERADATA_EXT`
- Vertica: `AWS_VERTICA_EXT`

Verwenden von benutzerdefinierten Bibliotheken für AWS SCT Erweiterungspakete

In einigen Fällen AWS SCT können Quelldatenbank-Features nicht in äquivalente Features in Ihrer Zieldatenbank konvertiert werden. Das entsprechende AWS SCT Erweiterungspaket enthält benutzerdefinierte Bibliotheken, die einige Funktionen der Quelldatenbank in Ihrer Zieldatenbank emulieren.

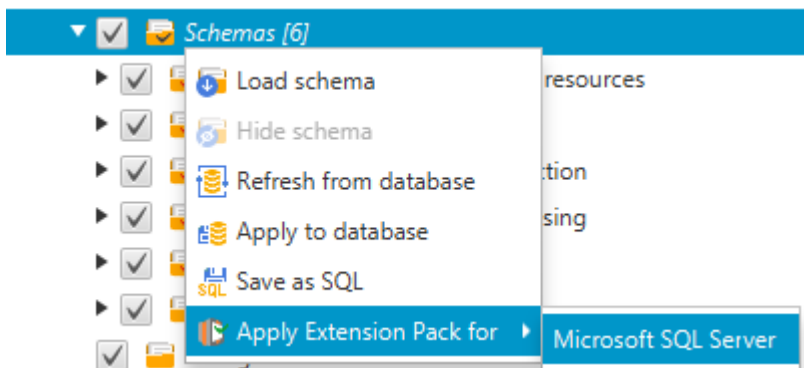
Wenn Sie eine Transaktionsdatenbank konvertieren, finden Sie weitere Informationen unter [Verwenden Sie die AWS Lambda Funktionen aus dem AWS SCT Erweiterungspaket](#)

Anwenden des Erweiterungspakets

Sie können das AWS SCT Erweiterungspaket mithilfe des Erweiterungspaket-Assistenten anwenden oder wenn Sie den konvertierten Code auf Ihre Zieldatenbank anwenden.

Um das Erweiterungspaket mit dem Erweiterungspaket-Assistenten anzuwenden

1. Öffnen Sie in der Zieldatenbankstruktur das Kontextmenü (Rechtsklick), wählen Sie Erweiterungspaket anwenden für und wählen Sie dann Ihre Quelldatenbankplattform aus. AWS Schema Conversion Tool



Der Assistent des Erweiterungspakets wird angezeigt.

2. Lesen Sie die Seite Welcome und wählen Sie dann Next aus.
3. Gehen Sie auf der Seite mit den AWS Profileinstellungen wie folgt vor:
 - Wenn Sie das Erweiterungspaketschema lediglich erneut installieren, wählen Sie Skip this step for now und Next aus. Die Option „Diesen Schritt vorerst überspringen“ ist nur für OLTP-Datenbanken (Online Transaction Processing) verfügbar.

- Wenn Sie eine neue Bibliothek hochladen, geben Sie die Anmeldeinformationen ein, um eine Verbindung zu Ihrer Bibliothek herzustellen. AWS-Konto Verwenden Sie diesen Schritt nur, wenn Sie OLAP-Datenbanken oder ETL-Skripts konvertieren. Sie können Ihre AWS Command Line Interface (AWS CLI) -Anmeldeinformationen verwenden, wenn Sie die AWS CLI installiert haben. Sie können auch die Anmeldeinformationen verwenden, die Sie zuvor in einem Profil in den globalen Anwendungseinstellungen gespeichert und mit dem Projekt verknüpft haben. Wählen Sie bei Bedarf Navigate to Global Settings aus, um Ihr AWS SCT Projekt zu konfigurieren oder einem anderen Profil zuzuordnen. Weitere Informationen finden Sie unter [Speichern von AWS Serviceprofilen in AWS SCT](#).
4. Wenn Sie eine neue Bibliothek hochladen, wählen Sie auf der Seite zum Hochladen der Bibliothek die Option Ich muss eine Bibliothek hochladen. Verwenden Sie diesen Schritt nur, wenn Sie OLAP-Datenbanken oder ETL-Skripts konvertieren. Geben Sie als Nächstes den Amazon S3 S3-Pfad an und wählen Sie dann Bibliothek auf S3 hochladen.

Wenn Sie die Bibliothek bereits hochgeladen haben, wählen Sie Ich habe bereits Bibliotheken hochgeladen und verwenden Sie meinen vorhandenen S3-Bucket auf der Seite zum Hochladen der Bibliothek. Geben Sie als Nächstes den Amazon S3-Pfad an.

Klicken Sie anschließend auf Next.

5. Wählen Sie auf der Seite Funktionsemulation die Option Erweiterungspaket erstellen aus. Es werden Nachrichten zum Status der Operationen des Erweiterungspakets angezeigt.

Wenn Sie fertig sind, wählen Sie Finish aus.

Um das Erweiterungspaket beim Anwenden des konvertierten Codes anzuwenden

1. Geben Sie den Amazon S3 S3-Bucket in Ihrem AWS Serviceprofil an. Verwenden Sie diesen Schritt nur, wenn Sie OLAP-Datenbanken oder ETL-Skripts konvertieren. Weitere Informationen finden Sie unter [Speichern von AWS Serviceprofilen in AWS SCT](#).

Stellen Sie sicher, dass Ihre Amazon S3 S3-Bucket-Richtlinie die folgenden Berechtigungen umfasst:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
    "Action": ["s3:ListBucket"],
    "Resource": ["*"]
  },
  {
    "Effect": "Allow",
    "Action": ["s3:PutObject"],
    "Resource": ["*"]
  },
  {
    "Effect": "Allow",
    "Action": ["iam:SimulatePrincipalPolicy"],
    "Resource": ["*"]
  },
  {
    "Effect": "Allow",
    "Action": ["iam:GetUser"],
    "Resource": ["arn:aws:iam::111122223333:user/DataExtractionAgentName"]
  }
]
```

Ersetzen Sie im vorherigen Beispiel *111122223333:user/ DataExtractionAgentName* durch den Namen Ihres IAM-Benutzers.

2. Konvertieren Sie Ihre Data Warehouse-Quellschemas. Weitere Informationen finden Sie unter [Konvertierung von Data Warehouse-Schemas nach Amazon Redshift](#).
3. Wählen Sie im rechten Bereich das konvertierte Schema aus.
4. Öffnen Sie das Kontextmenü (rechte Maustaste) für das Schema-Element und wählen Sie Apply to database aus.
5. AWS SCT generiert Erweiterungspakete mit den erforderlichen Komponenten und fügt das `aws_<database_engine_name>_ext` Schema in die Zielstruktur ein. AWS SCT Wendet als Nächstes den konvertierten Code und das Erweiterungspaket-Schema auf Ihr Ziel-Data Warehouse an.

Wenn Sie eine Kombination aus Amazon Redshift und AWS Glue als Zieldatenbankplattform verwenden, AWS SCT fügt dem Erweiterungspaket ein zusätzliches Schema hinzu.

Verwenden Sie die AWS Lambda Funktionen aus dem AWS SCT Erweiterungspaket

AWS SCT bietet ein Erweiterungspaket, das Lambda-Funktionen für E-Mail, Jobplanung und andere Funktionen für Datenbanken enthält, die auf Amazon EC2 gehostet werden.

Verwendung von AWS Lambda Funktionen zur Emulierung von Datenbankfunktionen

In einigen Fällen können Datenbankfunktionen nicht in äquivalente Amazon RDS-Funktionen konvertiert werden. Beispielsweise sendet Oracle E-Mail-Aufrufe mit UTL_SMTP, während Microsoft SQL Server möglicherweise einen Aufgabenplaner verwendet. Wenn Sie eine Datenbank auf Amazon EC2 hosten und selbst verwalten, können Sie diese Funktionen emulieren, indem Sie sie durch Dienste ersetzen AWS .

Der AWS SCT Erweiterungspaket-Assistent hilft Ihnen bei der Installation, Erstellung und Konfiguration von Lambda-Funktionen, um E-Mail, Jobplanung und andere Funktionen zu emulieren.

Anwenden des Erweiterungspakets zur Unterstützung von Lambda-Funktionen

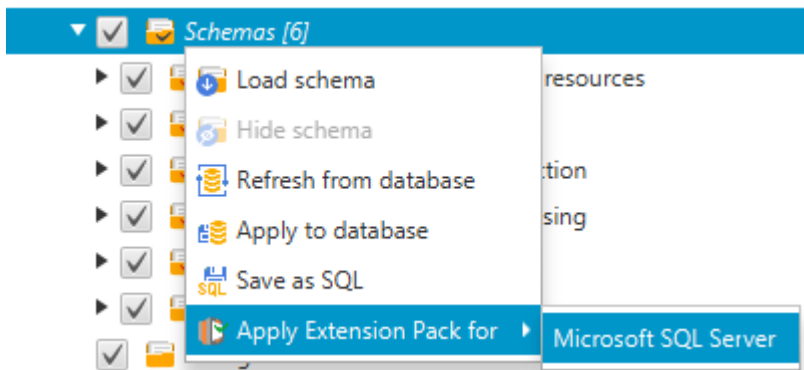
Sie können das Erweiterungspaket zur Unterstützung von Lambda-Funktionen mithilfe des Erweiterungspaket-Assistenten anwenden oder wenn Sie den konvertierten Code auf Ihre Zieldatenbank anwenden.

Important

Die Funktionen zur AWS Service-Emulation werden nur für Datenbanken unterstützt, die auf Amazon EC2 installiert und selbst verwaltet werden. Installieren Sie die Service-Emulationsfunktionen nicht, wenn sich Ihre Zieldatenbank auf einer Amazon RDS-DB-Instance befindet.

Um das Erweiterungspaket mithilfe des Erweiterungspaket-Assistenten anzuwenden

1. Öffnen Sie in der Zieldatenbankstruktur das Kontextmenü (Rechtsklick), wählen Sie Erweiterungspaket anwenden für und wählen Sie dann Ihre Quelldatenbankplattform aus. AWS Schema Conversion Tool



Der Assistent des Erweiterungspakets wird angezeigt.

2. Lesen Sie die Seite Welcome und wählen Sie dann Next aus.
3. Gehen Sie auf der Seite mit den AWS Profileinstellungen wie folgt vor:
 - Wenn Sie das Erweiterungspaketschema lediglich erneut installieren, wählen Sie Skip this step for now und Next aus.
 - Wenn Sie AWS Dienste installieren, geben Sie die Anmeldeinformationen ein, um eine Verbindung zu Ihrem herzustellen AWS-Konto. Sie können Ihre AWS CLI Anmeldeinformationen verwenden, wenn Sie die AWS CLI installiert haben. Sie können auch die Anmeldeinformationen verwenden, die Sie zuvor in einem Profil in den globalen Anwendungseinstellungen gespeichert und mit dem Projekt verknüpft haben. Wählen Sie bei Bedarf Navigate to Project Settings aus, um ein anderes Profil mit dem Projekt zu verknüpfen. Wählen Sie bei Bedarf Global Settings aus, um ein neues Profil zu erstellen. Weitere Informationen finden Sie unter [Speichern von AWS Serviceprofilen in AWS SCT](#).
4. Führen Sie auf der Seite Email Sending Service folgende Schritte aus:
 - Wenn Sie das Erweiterungspaketschema lediglich erneut installieren, wählen Sie Skip this step for now und Next aus.
 - Wenn Sie AWS Dienste installieren und bereits über eine Lambda-Funktion verfügen, können Sie diese bereitstellen. Andernfalls erstellt der Assistent eine für Sie. Klicken Sie anschließend auf Next.
5. Führen Sie auf der Seite Job Emulation Service folgende Schritte aus:
 - Wenn Sie das Erweiterungspaketschema lediglich erneut installieren, wählen Sie Skip this step for now und Next aus.

- Wenn Sie AWS Dienste installieren und bereits über eine Lambda-Funktion verfügen, können Sie diese bereitstellen. Andernfalls erstellt der Assistent eine für Sie. Klicken Sie anschließend auf Next.
6. Wählen Sie auf der Seite Funktionsemulation die Option Erweiterungspaket erstellen aus. Es werden Nachrichten zum Status der Operationen des Erweiterungspakets angezeigt.

Wenn Sie fertig sind, wählen Sie Finish aus.

Note

Um ein Erweiterungspaket zu aktualisieren und die alten Erweiterungspaket-Komponenten zu überschreiben, stellen Sie sicher, dass Sie die neueste Version von verwenden. AWS SCT
Weitere Informationen finden Sie unter [Installation, Überprüfung und Aktualisierung AWS SCT](#).

Funktionen für das AWS SCT Erweiterungspaket konfigurieren

Das Erweiterungspaket enthält Funktionen, die Sie vor der Verwendung konfigurieren müssen. Die Konstante `CONVERSION_LANG` definiert die Sprache, die das Service Pack verwendet. Die Funktionen sind für Englisch und Deutsch verfügbar.

Um die Sprache auf Englisch oder Deutsch einzustellen, nehmen Sie die folgende Änderung im Funktionscode vor. Suchen Sie die folgende Konstantendeklaration:

```
CONVERSION_LANG CONSTANT VARCHAR := '';
```

Um auf Englisch `CONVERSION_LANG` zu setzen, ändern Sie die Zeile wie folgt:

```
CONVERSION_LANG CONSTANT VARCHAR := 'English';
```

Um die Einstellung `CONVERSION_LANG` auf Englisch einzustellen, ändern Sie die Zeile wie folgt:

```
CONVERSION_LANG CONSTANT VARCHAR := 'Deutsch';
```

Stellen Sie diese Einstellung für die folgenden Funktionen ein:

- `aws_sqlserver_ext.conv_datetime_to_string`
- `aws_sqlserver_ext.conv_date_to_string`
- `aws_sqlserver_ext.conv_string_to_date`
- `aws_sqlserver_ext.conv_string_to_datetime`
- `aws_sqlserver_ext.conv_string_to_datetime`
- `aws_sqlserver_ext.parse_to_date`
- `aws_sqlserver_ext.parse_to_datetime`
- `aws_sqlserver_ext.parse_to_time`

Bewährte Methoden für AWS SCT

Hier finden Sie Informationen zu bewährten Methoden und Optionen für die Nutzung des AWS Schema Conversion Tool (AWS SCT).

Konfigurieren zusätzlichen

Um große Datenbankschemas zu konvertieren, z. B. eine Datenbank mit 3.500 gespeicherten Prozeduren, können Sie die Speichermenge konfigurieren, die dem zur Verfügung steht AWS Schema Conversion Tool.

Um die Menge an Speicherplatz zu ändern, die AWS SCT verbraucht wird

1. Wählen Sie im Menü Einstellungen die Option Globale Einstellungen und dann JVM-Optionen.
2. Wählen Sie Konfigurationsdatei bearbeiten und wählen Sie den Texteditor, um die Konfigurationsdatei zu öffnen.
3. Bearbeiten Sie den `JavaOptions`-Abschnitt, um den minimalen und maximalen verfügbaren Speicher festzulegen. Im folgenden Beispiel wird das Minimum auf vier GB und das Maximum auf 40 GB festgelegt.

```
[JavaOptions]
-Xmx40960M
-Xms4096M
```

Wir empfehlen, die Mindestkapazität auf mindestens vier GB einzustellen.

4. Speichern Sie die Konfigurationsdatei, wählen Sie OK und starten Sie neu, AWS SCT um die Änderungen zu übernehmen.

Konfiguration des Standard-Projektordners

AWS SCT verwendet den Projektordner zum Speichern der Projektdateien, zum Speichern von Bewertungsberichten und zum Speichern von konvertiertem Code. AWS SCT speichert standardmäßig alle Dateien im Anwendungsordner. Sie können einen anderen Ordner als Standard-Projektordner angeben.

Um den Standard-Projektordner zu ändern

1. Wählen Sie im Menü Einstellungen die Option Globale Einstellungen und dann Dateipfad aus.
2. Geben Sie unter Standard-Projektdateipfad den Pfad zum Standard-Projektordner ein.
3. Wählen Sie Übernehmen und anschließend OK aus.

Erhöhung der Datenmigration

Um große Datensätze zu migrieren, z. B. eine Reihe von Tabellen mit mehr als 1 TB Daten, möchten Sie möglicherweise die Migrationsgeschwindigkeit erhöhen. Wenn Sie Datenextraktionsagenten verwenden, hängt die Geschwindigkeit der Datenmigrationen von verschiedenen Faktoren ab. Zu diesen Faktoren gehören die Anzahl der Slices in Ihrem Amazon Redshift-Zielcluster, die Größe einer Chunk-Datei in Ihrer Migrationsaufgabe, der verfügbare Arbeitsspeicher auf dem PC, auf dem Sie Ihre Datenextraktionsagenten ausführen, und so weiter.

Um die Geschwindigkeit der Datenmigration zu erhöhen, empfehlen wir, mehrere Testmigrationssitzungen mit kleinen Datensätzen Ihrer Produktionsdaten durchzuführen. Außerdem empfehlen wir, dass Sie Ihre Datenextraktionsagenten auf einem PC mit einer SSD mit einer Größe von mindestens 500 GB ausführen. Ändern Sie während dieser Testsitzungen verschiedene Migrationsparameter und überwachen Sie Ihre Festplattenauslastung, um herauszufinden, welche Konfiguration die maximale Datenmigrationsgeschwindigkeit gewährleistet. Verwenden Sie dann diese Konfiguration, um Ihren gesamten Datensatz zu migrieren.

Erhöhung der Protokollierungsinformationen

Sie können die Protokollierungsinformationen erhöhen, die AWS SCT bei der Konvertierung Ihrer Datenbanken, Skripts und Anwendungs-SQL erzeugt werden. Eine Erhöhung der Protokollierungsinformationen kann zwar die Konvertierung verlangsamen, aber die Änderungen können Ihnen helfen, dem AWS Support zuverlässige Informationen zur Verfügung zu stellen, falls Fehler auftreten.

AWS SCT speichert Logs in Ihrer lokalen Umgebung. Sie können diese Protokolldateien einsehen und sie zur Fehlerbehebung an den AWS Support oder AWS SCT Entwickler weitergeben.

Um die Logging-Einstellungen zu ändern

1. Wählen Sie im Menü Einstellungen die Option Globale Einstellungen und dann Protokollierung aus.

2. Geben Sie unter Protokollordnerpfad den Ordner zum Speichern von Protokollen über die Benutzeroberfläche ein.
3. Geben Sie als Pfad zum Konsolenprotokollordner den Ordner ein, in dem die Protokolle der AWS SCT Befehlszeilenschnittstelle (CLI) gespeichert werden sollen.
4. Geben Sie für Maximale Protokolldateigröße (MB) die Größe einer einzelnen Protokolldatei in MB ein. Nachdem Ihre Datei dieses Limit erreicht hat, AWS SCT wird eine neue Protokolldatei erstellt.
5. Geben Sie für Maximale Anzahl von Protokolldateien die Anzahl der zu speichernden Protokolldateien ein. Wenn die Anzahl der Protokolldateien im Ordner diesen Grenzwert erreicht hat, AWS SCT wird die älteste Protokolldatei gelöscht.
6. Geben Sie als Downloadpfad für das Extraktoren-Protokoll den Ordner ein, in dem die Protokolle der Datenextraktionsagenten gespeichert werden.
7. Geben Sie für den Protokollpfad des Cassandra-Extraktors den Ordner ein, in dem die Protokolle der Apache Cassandra-Datenextraktionsagenten gespeichert werden.
8. Wählen Sie Vor dem Laden nach einem Pfad fragen aus, um sicherzustellen, dass jedes Mal AWS SCT gefragt wird, wo die Protokolle gespeichert werden sollen, wenn Sie Datenextraktionsagenten verwenden.
9. Wählen Sie für den Debug-Modus True. Verwenden Sie diese Option, um zusätzliche Informationen zu protokollieren, wenn AWS SCT Standardprotokolle keine Probleme enthalten.
10. Wählen Sie wichtige Anwendungsmodule aus, um die Protokollierungsinformationen zu erweitern. Sie können die Protokollierungsinformationen für die folgenden Anwendungsmodule erhöhen:
 - Allgemeines
 - Loader
 - Parser
 - Drucker
 - Resolver
 - Telemetrie
 - Konverter
 - Geben Sie Mapping ein
 - Benutzeroberfläche
 - Kontrolleur

- Schema vergleichen
- Rechenzentrum klonen
- Anwendungsanalysator

Wählen Sie für die nachstehenden Optionen aus:

- Trace — Die detailliertesten Informationen.
- Debug — Detaillierte Informationen zum Ablauf des Systems.
- Info — Laufzeitereignisse wie Start oder Herunterfahren.
- Warnung — Verwendung veralteter APIs, schlechte API-Verwendung, andere Laufzeitsituationen, die unerwünscht oder unerwartet sind.
- Fehler — Laufzeitfehler oder unerwartete Bedingungen.
- Kritisch — Fehler, die zum Herunterfahren der Anwendung führen.
- Obligatorisch — Die höchstmögliche Fehlerquote.

AWS SCT legt nach dem Aktivieren des Debug-Modus standardmäßig die Informationsprotokollierungsebene für alle Anwendungsmodule fest.

Um beispielsweise bei wichtigen Problembereichen bei der Konvertierung zu helfen, stellen Sie Parser, Typzuordnung und Benutzeroberfläche auf Trace ein.

Wenn die Informationen für das Dateisystem, in dem die Protokolle gestreamt werden, zu umfangreich werden, wechseln Sie zu einem Speicherort, an dem ausreichend Speicherplatz zum Erfassen von Protokollen vorhanden ist.

Um Protokolle an den AWS Support zu übertragen, wechseln Sie in das Verzeichnis, in dem die Protokolle gespeichert sind, und komprimieren Sie alle Dateien in einer überschaubaren ZIP-Datei. Laden Sie dann die ZIP-Datei mit Ihrer Support-Anfrage hoch. Wenn die erste Analyse abgeschlossen ist und die laufende Entwicklung fortgesetzt wird, setzen Sie den Debug-Modus auf false zurück, um die ausführliche Protokollierung zu vermeiden. Erhöhen Sie dann die Konvertierungsgeschwindigkeit.

 Tip

Um die Protokollgröße zu verwalten und Probleme mit der Berichterstattung zu vereinfachen, entfernen Sie die Protokolle oder verschieben Sie sie nach einer erfolgreichen Konvertierung an einen anderen Ort. Durch diese Aufgabe wird sichergestellt, dass nur die relevanten Fehler und Informationen an den AWS Support übertragen werden, und verhindert, dass sich das Log-Dateisystem füllt.

Beheben von Problemen mit Problemen mit AWS SCT

Im Folgenden finden Sie Informationen zur Fehlersuche für das AWS Schema Conversion Tool (AWS SCT).

Objekte können nicht aus einer Oracle-Quelldatenbank geladen werden.

Wenn Sie versuchen, ein Schema aus einer Oracle-Datenbank zu laden, tritt möglicherweise einer der folgenden Fehler auf.

```
Cannot load objects tree.
```

```
ORA-00942: table or view does not exist
```

Diese Fehler treten auf, da der Benutzer, dessen ID Sie zum Herstellen einer Verbindung mit der Oracle-Datenbank verwendeten, nicht über die erforderlichen Berechtigungen zum Lesen des Schemas verfügt, wie von AWS SCT gefordert.

Sie können das Problem lösen, indem Sie dem Benutzer die `select_catalog_role`-Berechtigung gewähren und außerdem die Berechtigung für jedes Wörterbuch in der Datenbank. Diese Berechtigungen bieten den schreibgeschützten Zugriff auf die Ansichten und Systemtabellen, der für AWS SCT erforderlich ist. Im folgenden Beispiel wird eine Benutzer-ID mit dem Namen `min_privs` erstellt und dem Benutzer mit dieser ID werden die Minimalberechtigungen gewährt, die für die Konvertierung eines Schemas von einer Oracle-Quelldatenbank erforderlich sind.

```
create user min_privs identified by min_privs;  
grant connect to min_privs;  
grant select_catalog_role to min_privs;  
grant select any dictionary to min_privs;
```

Warnmeldung des Bewertungsberichts

Um die Komplexität der Konvertierung zu einer anderen Datenbank-Engine beurteilen zu können, AWS SCT ist Zugriff auf Objekte in Ihrer Quelldatenbank erforderlich. Wenn AWS

SCT beim Scannen Probleme auftreten und keine Bewertung durchgeführt werden kann, wird eine Warnmeldung ausgegeben. Diese Meldung weist darauf hin, dass der gesamte Konversionsprozentsatz reduziert wurde. Im Folgenden sind die Gründe aufgeführt, warum beim Scannen ProblemeAWS SCT auftreten können:

- Ihr Datenbankbenutzer hat nicht Zugriff auf alle benötigten Objekte. Weitere Informationen zu denAWS SCT erforderlichen Sicherheitsberechtigungen und -berechtigungen für Ihre Datenbank finden Sie im Abschnitt[Quellen für AWS SCT](#) zur entsprechenden Quelldatenbank in diesem Handbuch.
- Ein im Schema zitiertes Objekt ist in der Datenbank nicht mehr vorhanden. Um das Problem zu lösen, können Sie eine Verbindung mit SYSDBA-Berechtigungen herstellen und überprüfen, ob das Objekt in der Datenbank vorhanden ist.
- SCT versucht, ein verschlüsseltes Objekt zu bewerten.

AWS SCT CLI-Referenz

In diesem Abschnitt werden die ersten Schritte mit der AWS SCT Befehlszeilenschnittstelle (CLI) beschrieben. Außerdem enthält dieser Abschnitt Informationen zu den Tastenbefehlen und Verwendungsmodi. Eine vollständige Referenz der AWS SCT CLI-Befehle finden Sie unter [Referenzmaterial](#).

Themen

- [Voraussetzungen für die Verwendung der AWS SCT Befehlszeilenschnittstelle](#)
- [AWS SCT Interaktiver CLI-Modus](#)
- [AWS SCT CLI-Szenarien abrufen](#)
- [AWS SCT CLI-Szenarien bearbeiten](#)
- [AWS SCT CLI-Skriptmodus](#)
- [AWS SCT CLI-Referenzmaterial](#)

Voraussetzungen für die Verwendung der AWS SCT Befehlszeilenschnittstelle

Laden Sie die neueste Version von Amazon Corretto 11 herunter und installieren Sie sie. Weitere Informationen finden Sie unter [Downloads für Amazon Corretto 11](#) im Amazon Corretto 11-Benutzerhandbuch.

Laden Sie die neueste Version von herunter und installieren Sie sie. AWS SCT Weitere Informationen finden Sie unter [Wird installiert AWS SCT](#).

AWS SCT Interaktiver CLI-Modus

Sie können die AWS SCT Befehlszeilenschnittstelle im interaktiven Modus verwenden. In diesem Modus geben Sie Befehle nacheinander in die Konsole ein. In diesem interaktiven Modus können Sie mehr über CLI-Befehle erfahren oder die am häufigsten verwendeten CLI-Szenarien herunterladen.

Um Ihr Quelldatenbankschema zu konvertieren AWS SCT, führen Sie einen Sequenzvorgang aus: Erstellen Sie ein neues Projekt, stellen Sie eine Verbindung zu Quell- und Zieldatenbanken her, erstellen Sie Zuordnungsregeln und konvertieren Sie Datenbankobjekte. Da dieser Workflow

komplex sein kann, empfehlen wir die Verwendung von Skripten im AWS SCT CLI-Modus. Weitere Informationen finden Sie unter [Skriptmodus](#).

Sie können die AWS SCT CLI-Befehle im app Ordner Ihres AWS SCT Installationspfads ausführen. In Windows lautet der Standardinstallationspfad `C:\Program Files\AWS Schema Conversion Tool\`. Stellen Sie sicher, dass dieser Ordner die `AWSSchemaConversionToolBatch.jar` Datei enthält.

Verwenden Sie den folgenden Befehl, um in den interaktiven AWS SCT CLI-Modus zu wechseln, nachdem Sie die Voraussetzungen erfüllt haben.

```
java -jar AWSSchemaConversionToolBatch.jar -type interactive
```

Jetzt können Sie AWS SCT CLI-Befehle ausführen. Stellen Sie sicher, dass Sie Ihre Befehle mit `/` in einer neuen Zeile beenden. Stellen Sie außerdem sicher, dass Sie vor und nach den Werten der Befehlsparameter gerade einfache Anführungszeichen (`'`) verwenden.

Note

Wenn der vorherige Befehl zurückkehrt `Unexpected error`, versuchen Sie Folgendes:

```
java -Djdk.jar.maxSignatureFileSize=20000000 -jar  
AWSSchemaConversionToolBatch.jar
```

Führen Sie den folgenden Befehl aus, um die Liste der verfügbaren Befehle im interaktiven AWS SCT CLI-Modus anzuzeigen.

```
help  
/
```

Verwenden Sie den folgenden Befehl, um Informationen zu einem AWS SCT CLI-Befehl anzuzeigen.

```
help -command: 'command_name'  
/
```

Ersetzen Sie im vorherigen Beispiel *command_name* durch den Namen eines Befehls.

Verwenden Sie den folgenden Befehl, um Informationen zu den Parametern eines AWS SCT CLI-Befehls anzuzeigen.

```
help -command: 'command_name' -parameters: 'parameters_list'  
/
```

Ersetzen Sie im vorherigen Beispiel *command_name* durch den Namen eines Befehls. Ersetzen Sie dann *parameters_list* durch eine Liste von Parameternamen, die durch ein Komma getrennt sind.

Verwenden Sie den folgenden Befehl, um ein Skript aus einer Datei im interaktiven AWS SCT CLI-Modus auszuführen.

```
ExecuteFile -file: 'file_path'  
/
```

Ersetzen Sie im vorherigen Beispiel *file_path* durch den Pfad zu Ihrer Datei mit einem Skript. Stellen Sie sicher, dass Ihre Datei eine `.scts` Erweiterung hat.

Führen Sie den `quit` Befehl aus, um den interaktiven AWS SCT CLI-Modus zu beenden.

Beispiele

Im folgenden Beispiel werden die Informationen über den `Convert` Befehl angezeigt.

```
help -command: 'Convert'  
/
```

Im folgenden Beispiel werden die Informationen zu zwei Parametern des `Convert` Befehls angezeigt.

```
help -command: 'Convert' -parameters: 'filter, treePath'  
/
```

AWS SCT CLI-Szenarien abrufen

Um die am häufigsten verwendeten AWS SCT Szenarien abzurufen, können Sie den `GetCliScenario` Befehl verwenden. Sie können diesen Befehl im interaktiven Modus ausführen

und dann die heruntergeladenen Vorlagen bearbeiten. Verwenden Sie die bearbeiteten Dateien im Skriptmodus.

Der `GetCliScenario` Befehl speichert die ausgewählte Vorlage oder alle verfügbaren Vorlagen im angegebenen Verzeichnis. Die Vorlage enthält den vollständigen Befehlssatz zum Ausführen eines Skripts. Stellen Sie sicher, dass Sie die Dateipfade, Datenbankanmeldeinformationen, Objektnamen und andere Daten in diesen Vorlagen bearbeiten. Stellen Sie außerdem sicher, dass Sie die Befehle entfernen, die Sie nicht verwenden, und fügen Sie dem Skript bei Bedarf neue Befehle hinzu.

Um den `GetCliScenario` Befehl auszuführen, erfüllen Sie die Voraussetzungen und wechseln Sie in den interaktiven AWS SCT CLI-Modus. Weitere Informationen finden Sie unter [interaktiver Modus](#).

Verwenden Sie als Nächstes die folgende Syntax, um den `GetCliScenario` Befehl auszuführen und die AWS SCT Szenarien abzurufen.

```
GetCliScenario -type: 'template_type' -directory: 'file_path'
/
```

Ersetzen Sie im vorherigen Beispiel *template_type* durch einen der Vorlagentypen aus der folgenden Tabelle. Ersetzen Sie anschließend *file_path* durch den Pfad des Ordners, in den Sie Skripts herunterladen möchten. Stellen Sie sicher, dass Sie auf diesen Ordner zugreifen AWS SCT können, ohne Administratorrechte anzufordern. Stellen Sie außerdem sicher, dass Sie vor und nach den Werten der Befehlsparameter gerade einfache Anführungszeichen (') verwenden.

Um alle AWS SCT CLI-Vorlagen herunterzuladen, führen Sie den vorherigen Befehl ohne die `-type` Option aus.

Die folgende Tabelle enthält die Arten von AWS SCT CLI-Vorlagen, die Sie herunterladen können. Für jede Vorlage enthält die Tabelle den Dateinamen und die Beschreibung der Operationen, die Sie mit dem Skript ausführen können.

Typ der Vorlage	Dateiname	Beschreibung
BTEQ ScriptConversion	BTEQScriptConversionTemplate.scts	Konvertiert Teradata Basic Teradata Query (BTEQ), FastExport, und MultiLoad Skripten in Amazon Redshift FastLoad RSQL. Weitere

Typ der Vorlage	Dateiname	Beschreibung
		Informationen finden Sie unter ETL-Prozesse konvertieren .
ConversionApply	ConversionTemplate.scts	Konvertiert Quelldatenbankschemas und wendet den konvertierten Code auf die Zieldatenbank an. Speichert optional den konvertierten Code als SQL-Skript und speichert den Bewertungsbericht. Weitere Informationen finden Sie unter Datenbankschemas konvertieren .
GenericAppConversion	GenericApplicationConversionTemplate.scts	Konvertiert mit dem generischen AWS SCT Anwendungskonverter in Ihre Anwendungen eingebetteten SQL-Code. Weitere Informationen finden Sie unter Konvertieren von SQL-Code in Ihren Anwendungen .
HadoopMigration	HadoopMigrationTemplate.scts	Migriert Ihren lokalen Hadoop-Cluster zu Amazon EMR. Weitere Informationen finden Sie unter Verwendung von Apache Hadoop als Quelle für AWS SCT .

Typ der Vorlage	Dateiname	Beschreibung
HadoopResumeMigration	HadoopResumeMigrationTemplate.scts	Setzt eine unterbrochene Migration Ihres lokalen Hadoop-Clusters zu Amazon EMR fort. Weitere Informationen finden Sie unter Verwendung von Apache Hadoop als Quelle für AWS SCT .
Informatica	InformaticaConversionTemplate.scts	Konvertiert SQL-Code, der in Ihre Informatica-Skripts zum Extrahieren, Transformieren und Laden (ETL) eingebettet ist. Konfiguriert Verbindungen zu Ihren Quell- und Zieldatenbanken in Ihren ETL-Skripten und speichert die konvertierten Skripts nach der Konvertierung. Weitere Informationen finden Sie unter Konvertierung von Informatica ETL-Skripten .
LanguageSpecificAppConversion	LanguageSpecificAppConversionTemplate.scts	Konvertiert mit dem Anwendungskonverter SQL-Code, der in Ihre C#-, C++-, Java- und Pro*C-Anwendungen eingebettet ist. AWS SCT Weitere Informationen finden Sie unter Anwendungs-SQL konvertieren .

Typ der Vorlage	Dateiname	Beschreibung
OozieConversion	OozieConversionTemplate.scts	Konvertiert Ihre Apache Oozie-Workflows in. AWS Step Functions Weitere Informationen finden Sie unter Verwendung von Apache Oozie als Quelle für AWS SCT .
RedshiftAgent	DWHDataMigrationTemplate.scts	Konvertiert Quell-Data Warehouse-Schemas und wendet den konvertierten Code auf die Amazon Redshift Redshift-Zieldatenbank an. Registriert dann einen Datenextraktionsagenten, erstellt und startet eine Datenmigrationsaufgabe. Weitere Informationen finden Sie unter Migration von einem Data Warehouse zu Amazon Redshift .
ReportCreation	ReportCreationTemplate.scts	Erstellt einen Datenbank migrationsbericht für mehrere Quelldatenbankschemas. Speichert diesen Bericht dann als CSV- oder PDF-Datei. Weitere Informationen finden Sie unter Berichte über die Migrationsbewertung .

Typ der Vorlage	Dateiname	Beschreibung
SQL ScriptConversion	SQLScriptConversionTemplate.scts	Konvertiert SQL*Plus- oder TSQL-Skripten nach PL/SQL und speichert konvertierte Skripten. Speichert außerdem einen Bewertungsbericht.

Nachdem Sie die AWS SCT CLI-Vorlage heruntergeladen haben, konfigurieren Sie das Skript mit dem Texteditor so, dass es in Ihren Quell- und Zieldatenbanken ausgeführt wird. Verwenden Sie als Nächstes den AWS SCT CLI-Skriptmodus, um Ihr Skript auszuführen. Weitere Informationen finden Sie unter [AWS SCT CLI-Skriptmodus](#).

Beispiele

Im folgenden Beispiel werden alle Vorlagen in den C:\SCT\Templates Ordner heruntergeladen.

```
GetCliScenario -directory: 'C:\SCT\Templates'  
/
```

Im folgenden Beispiel wird die Vorlage für den ConversionApply Vorgang in den C:\SCT\Templates Ordner heruntergeladen.

```
GetCliScenario -type: 'ConversionApply' -directory: 'C:\SCT\Templates'  
/
```

AWS SCT CLI-Szenarien bearbeiten

Nachdem Sie Szenariovorlagen heruntergeladen haben, konfigurieren Sie sie so, dass funktionierende Skripts verfügbar sind, die in Ihren Datenbanken ausgeführt werden können.

Stellen Sie bei allen Vorlagen sicher, dass Sie den Pfad zu den Treibern für Ihre Quell- und Zieldatenbanken angeben. Weitere Informationen finden Sie unter [Die erforderlichen Datenbanktreiber werden heruntergeladen](#).

Stellen Sie sicher, dass Sie die Datenbankanmeldedaten für Quell- und Zieldatenbanken angeben. Stellen Sie außerdem sicher, dass Sie Zuordnungsregeln einrichten, um ein Quell-Ziel-Paar für Ihr

Konvertierungsprojekt zu beschreiben. Weitere Informationen finden Sie unter [Mapping-Regeln erstellen](#).

Als Nächstes konfigurieren Sie den Umfang der auszuführenden Operationen. Sie können die Befehle entfernen, die Sie nicht verwenden, oder dem Skript neue Befehle hinzufügen.

Nehmen wir beispielsweise an, Sie planen, alle Schemas in Ihrer Oracle-Quelldatenbank nach PostgreSQL zu konvertieren. Anschließend planen Sie, Ihren Bewertungsbericht zur Datenbankmigration als PDF zu speichern und den konvertierten Code auf die Zieldatenbank anzuwenden. In diesem Fall können Sie die Vorlage für den `ConversionApply` Vorgang verwenden. Gehen Sie wie folgt vor, um Ihre AWS SCT CLI-Vorlage zu bearbeiten.

Um die AWS SCT CLI-Vorlage für den **ConversionApply** Vorgang zu bearbeiten

1. Öffnen Sie die Datei `ConversionTemplate.scts`, die Sie heruntergeladen haben. Weitere Informationen finden Sie unter [Beispiele](#).
2. Entfernen `CreateFilter` und konvertieren Sie `-filter-`, `ApplyToTarget -filter -`, `SaveTargetSQL -LbyStatement`, `SaveTargetSQ -` und `SaveReportCSV`-Operationen aus Ihrem Skript.
3. Geben Sie für `oracle_driver_file` im `SetGlobalSettings` Vorgang den Pfad zu Ihrem Oracle-Treiber ein. Geben Sie dann für `postgresql_driver_file` den Pfad zu Ihrem PostgreSQL-Treiber ein.

Wenn Sie andere Datenbank-Engines verwenden, verwenden Sie die entsprechenden Namen für die Einstellungen. Eine vollständige Liste der globalen Einstellungen, die Sie während des `SetGlobalSettings` Vorgangs festlegen können, finden Sie unter [Globale Einstellungsmatrix in der Referenzmaterial](#).

4. (Optional) Geben Sie für `CreateProject` den Namen Ihres Projekts und den Speicherort Ihrer lokalen Projektdatei ein. Wenn Sie sich dafür entscheiden, mit den Standardwerten fortzufahren, stellen Sie sicher, dass Dateien in dem `C:\temp` Ordner erstellt werden AWS SCT können, ohne Administratorrechte anzufordern.
5. Geben Sie für `AddSource` die IP-Adresse Ihres Quelldatenbankservers ein. Geben Sie außerdem den Benutzernamen, das Passwort und den Port ein, um eine Verbindung zu Ihrem Quelldatenbankserver herzustellen.
6. Geben Sie für `AddTarget` die IP-Adresse Ihres Zieldatenbankservers ein. Geben Sie außerdem den Benutzernamen, das Passwort und den Port ein, um eine Verbindung zu Ihrem Zieldatenbankserver herzustellen.
7. (Optional) Geben Sie für `AddServerMapping` die Quell- und Zieldatenbankobjekte ein, die Sie einer Zuordnungsregel hinzufügen möchten. Sie können `targetTreePath` Parameter

`sourceTreePath` und verwenden, um den Pfad zu den Datenbankobjekten anzugeben. Optional können Sie `sourceNamePath` und verwenden, `targetNamePath` um die Namen der Datenbankobjekte anzugeben. Weitere Informationen finden Sie unter Befehle zur Serverzuordnung in der [Referenzmaterial](#).

Die Standardwerte des `AddServerMapping` Vorgangs ordnen alle Quellschemas Ihrer Zieldatenbank zu.

- Speichern Sie die Datei und verwenden Sie dann den Skriptmodus, um sie auszuführen. Weitere Informationen finden Sie unter [Skriptmodus](#).

AWS SCT CLI-Skriptmodus

Nachdem Sie ein AWS SCT CLI-Skript erstellt oder eine Vorlage bearbeitet haben, können Sie es mit dem `RunSCTBatch` Befehl ausführen. Stellen Sie sicher, dass Sie Ihre Datei mit dem CLI-Skript als `.scts` Erweiterung speichern.

Sie können AWS SCT CLI-Skripte aus dem `app` Ordner Ihres AWS SCT Installationspfads ausführen. In Windows lautet der Standardinstallationspfad `C:\Program Files\AWS Schema Conversion Tool\`. Stellen Sie sicher, dass dieser Ordner die `RunSCTBatch.sh` Datei `RunSCTBatch.cmd` oder enthält. Außerdem sollte dieser Ordner die `AWSSchemaConversionToolBatch.jar` Datei enthalten.

Alternativ können Sie den Pfad zur `RunSCTBatch` Datei in der `PATH` Umgebungsvariablen Ihres Betriebssystems hinzufügen. Nachdem Sie die `PATH` Umgebungsvariable aktualisiert haben, können Sie AWS SCT CLI-Skripts von jedem Ordner aus ausführen.

Verwenden Sie den folgenden Befehl in Windows, um ein AWS SCT CLI-Skript auszuführen.

```
RunSCTBatch.cmd --pathtoscts "file_path"
```

Ersetzen Sie im vorherigen Beispiel *file_path* durch den Pfad zu Ihrer Datei mit einem Skript.

Verwenden Sie den folgenden Befehl unter Linux, um ein AWS SCT CLI-Skript auszuführen.

```
RunSCTBatch.sh --pathtoscts "file_path"
```

Ersetzen Sie im vorherigen Beispiel *file_path* durch den Pfad zu Ihrer Datei mit einem Skript.

Sie können in diesem Befehl optionale Parameter angeben, z. B. Datenbankmeldedaten, den Detaillierungsgrad in der Konsolenausgabe und andere. Weitere Informationen finden Sie in der Referenz zur AWS SCT Befehlszeilenschnittstelle unter [Referenzmaterial](#).

Beispiele

Im folgenden Beispiel wird das `ConversionTemplate.scts` Skript im `C:\SCT\Templates` Ordner ausgeführt. Sie können dieses Beispiel in Windows verwenden.

```
RunSCTBatch.cmd --pathtoscts "C:\SCT\Templates\ConversionTemplate.scts"
```

Im folgenden Beispiel wird das `ConversionTemplate.scts` Skript im `/home/user/SCT/Templates` Verzeichnis ausgeführt. Sie können dieses Beispiel unter Linux verwenden.

```
RunSCTBatch.sh --pathtoscts "/home/user/SCT/Templates/ConversionTemplate.scts"
```

AWS SCT CLI-Referenzmaterial

Referenzmaterial zur AWS Schema Conversion Tool Befehlszeilenschnittstelle (CLI) finden Sie in der folgenden Anleitung: [AWS Schema Conversion Tool CLI Reference](#).

Versionshinweise für AWS SCT

Dieser Abschnitt enthält Versionshinweise für AWS SCT, beginnend mit Version 1.0.640.

Versionshinweise für AWS SCT Build 675

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Cassandra	DynamoDB	Es wurde ein Fehler behoben, bei dem die Installation von Cassandra im Zielrechenzentrum fehlschlug.	Nein
DB2 LUW	PostgreSQL	DYNAMIC SQL: PREPARE-Anweisung: Auflösung und Konvertierung ohne dynamisches SQL.	Nein
DB2 LUW	PostgreSQL	Unterstützung für SPECIAL REGISTER hinzugefügt.	Nein
DB2 LUW	PostgreSQL	Aktualisierung des Erweiterungspakets	Nein
Hadoop	Amazon EMR	Unterstützung für die Verbindung zu einem Hadoop-Cluster über das rsa-sha2-Protokoll hinzugefügt.	Nein
Microsoft SQL Server	Amazon Redshift	Korrektur für den JDBC-Treiber, der TLS erzwingt, obwohl er nicht konfiguriert ist.	Nein
Netezza	Amazon Redshift	Unterstützung für die Konvertierung von Materialized Views hinzugefügt.	Nein
Oracle	Amazon Redshift	Unterstützung für rekursive Abfragen in Amazon Redshift hinzugefügt.	Ja
Oracle	PostgreSQL,	Die falsche Konvertierung des NUMBER-Datentyps wurde behoben.	Ja

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
	Aurora PostgreSQL		
Oracle	Amazon Redshift	Datenmigration. Automatische Partitionierung von Oracle. Die Ablaufzeit für den Wert für Tabellenfragmente wurde hinzugefügt. Die Ablaufzeit beträgt 72 Stunden. Wenn ein Ablauf eintritt, werden Datenfragmente neu erstellt, wenn eine Datenmigrationaufgabe erstellt wird.	Nein
Oracle	Amazon Redshift	SCT Data Extractor: Der Ansatz zum Hochladen von Daten auf Amazon Redshift wurde geändert. Standardmäßig erstellt der Extraktor keine Staging-Tabellen. Nachdem sich alle Datendateien im Amazon S3 S3-Bucket befinden, kopiert der Extraktor sie stattdessen mit einem einzigen COPY-Befehl in die Zieltabelle.	Nein
Oracle	Amazon Redshift	Migration des RAW-Datentyps zur VARBYTE-Spalte hinzugefügt.	Nein
Oracle	PostgreSQL, Aurora PostgreSQL	Konvertierung mehrerer Versionen	Nein
Oracle	PostgreSQL	Unterstützung für MERGE Statement in PostgreSQL 15.x hinzugefügt.	Ja
Oracle	PostgreSQL	Unterstützung für neue Funktionen für reguläre Ausdrücke in PostgreSQL 15.x hinzugefügt.	Ja

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Oracle	PostgreSQL, Aurora PostgreSQL	Die ON CONFLICT DO UPDATE-Anweisung wird ohne ausgeschlossenen Alias konvertiert.	Ja
Teradata	Amazon Redshift	Konvertierungsunterstützung für LEAD/LAG-Funktionen hinzugefügt.	Nein
Teradata	Amazon Redshift	Verbesserte Datentypumwandlung mit expliziter Angabe des Datenformats.	Nein
Teradata	Amazon Redshift	Verbesserte Konvertierung der AT 'TIME ZONE'-Klausel in Zeit-/Zeitstempelausdrücken.	Nein
Teradata	Amazon Redshift	AI-9996 bei Konvertierungsvorgängen mit MERGE-Anweisungen.	Nein

Versionshinweise für Build 674 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Alle	Alle	Verschiedene Fehlerkorrekturen und Leistungsverbesserungen	Teilweise (nur für unterstützte Quell- und Zielpaare)
Azure SQL/ Microsoft	Amazon Redshift	Die Meldung „AI 18066: Der Schemaname kann nicht konvertiert werden“, die Benutzer bei der	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
SQL Server		Schemabewertung/-konvertierung irreführte, wurde entfernt	
Azure SQL/ Microsoft SQL Server	Amazon RDS for MySQL// Amazon Aurora MySQL	Fehlerhafte Konvertierung des Verfahrens ohne Zuweisung eines Returncodes	Teilweise (Die Schemakonvertierung unterstützt Azure SQL derzeit nicht als Quelle)
Azure SQL/ Microsoft SQL Server	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	AI9997 für einige Fälle der Konvertierung von FOR XML PATH-Klauseln wurde behoben	Teilweise (Die Schemakonvertierung unterstützt Azure SQL derzeit nicht als Quelle)
Azure SQL/ Microsoft SQL Server	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Der Wert wird im Prozedur-/Funktionstext auf die ursprüngliche Skala gerundet	Teilweise (Die Schemakonvertierung unterstützt Azure SQL derzeit nicht als Quelle)

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Azure SQL/ Microsoft SQL Server	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Verschiedene Verbesserungen bei der Konvertierung von EXECUTE-Anweisungen	Teilweise (Die Schemakonvertierung unterstützt Azure SQL derzeit nicht als Quelle)
Azure SQL/ Microsoft SQL Server/ Azure Synapse	Amazon Redshift	Verbesserte Konvertierung der folgenden Anweisungen und Modi: <ul style="list-style-type: none"> • AUSNAHMEBLOCK • AUTOCOMMIT • NONATOMIC • GRUPPIERUNGSSATZ • CUBE • ROLLUP 	Nein
DB2 LUW	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Verschiedene Korrekturen beim Laden von SQL-Abfragen für Metadaten	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
DB2 LUW	Amazon RDS for PostgreSQL L// Amazon Aurora PostgreSQL	AI 9996 wird bei Triggern nicht erwartet	Nein
DB2 z/OS	Amazon RDS for PostgreSQL L// Amazon Aurora PostgreSQL	Analytische Funktion ROWNUMBER	Nein
DB2 z/OS	Amazon RDS for PostgreSQL L// Amazon Aurora PostgreSQL	Unterstützung für hexadezimale Zeichenkettenkonstanten	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Verschiedene Korrekturen beim Laden von SQL-Abfragen für Metadaten.	Nein
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	NÄCHSTER WERT FÜR die Unterstützung von Sequenzreferenzen	Nein
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Unterstützung der DB2_NUMBER_ROWS-Option der GET DIAGNOSTICS-Anweisung	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	GET DIAGNOSTICS Mehrere Anweisungen	Nein
DB2 z/OS	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Fehler bei der Konvertierung von FOR-Anweisungen wurden behoben.	Nein
Oracle	Amazon RDS for MySQL// Amazon Aurora MySQL	Ein Fehler wurde behoben, bei dem der Parameter knoten der Paketfunktion nicht definiert war.	Ja

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
Oracle	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Fehler in der Funktion AWS_ORACLE_EXT.NEXT_DAY des Erweiterungspakets wurden behoben	Ja
Oracle	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Verschiedene Fehler bei der Konvertierung „(+“ in den Outer Joins von Oracle wurden behoben	Ja
Oracle		Support der Kerberos-Authentifizierung	Nein
SAP ASE	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Ein Fehler beim Konvertieren von mehr als einem Bezeichner in der FROM-Klausel in der UPDATE-Anweisung wurde behoben	Nein

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	Verfügbarkeit bei der AWS DMS Schemakonvertierung
SAP ASE	Amazon RDS for PostgreSQL// Amazon Aurora PostgreSQL	Ein Fehler bei der Konvertierung mehrzeiliger Kommentare und Anweisungen wurde behoben	Nein
SAP ASE		Unterstützung für den Parameter ENCRYPT_PASSWORD beim Herstellen einer Verbindung hinzugefügt	Nein
Teradata	Amazon Redshift	Die Konvertierung der VOLATILE-Tabelle mit dem angegebenen Schemanamen wurde verbessert	Nein
Teradata	Amazon Redshift	Falsche Konvertierung WHERE-KLAUSEL in komplexem CTE	Nein
Teradata	Amazon Redshift	Unterstützung für den INTERVAL-Datentyp bei der Migration von Daten mit SCT-Datenextraktionsagenten wurde hinzugefügt.	Nein
Teradata BTEQ-Skripte	Amazon Redshift RSQL-Skripts	Falsche Ausgangsparameter für die Konvertierung in der von BTEQ ausgeführten Prozedur	Nein

Versionshinweise für Build 673 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Allgemeine Fehlerkorrekturen und Leistungsverbesserungen
Azure SQL/ Microsoft SQL Server	Aurora PostgreSQL / Amazon RDS PostgreSQL	Fehlerhafte Konvertierung von Funktionsaufrufen wurde behoben
Azure SQL/ Microsoft SQL Server	Aurora PostgreSQL / Amazon RDS PostgreSQL	Die Konvertierung der XML-Klausel wurde implementiert FOR
Azure SQL/ Microsoft SQL Server	Aurora PostgreSQL / Amazon RDS PostgreSQL	Konvertierung der FOR XML-Klausel mit falschem Alias.
Azure SQL/ Microsoft SQL Server	Aurora PostgreSQL / Amazon RDS PostgreSQL	Es wurde ein Fehler behoben, der AWS SCT auftrat, wenn EXECUTE Anweisungen, die eine Zeichenfolge mit Prozedurparametern ausführen, nicht konvertiert wurden.
Azure SQL/ Microsoft	Aurora PostgreSQL / Amazon RDS	Verbesserte Konvertierung von UPDATE-Anweisungen mit inneren Joins.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SQL Server	PostgreSQL	
Azure Synapse	Amazon Redshift	Die falsche Konvertierung der OBJECT_ID integrierten Funktion wurde behoben.
IBM DB2 für z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Die Konvertierung der folgenden Anweisungen und Objekte wurde implementiert: <ul style="list-style-type: none"> • DECLARE TEMPORARY TABLE statement • DROP TABLE statement • PKund UNIQUE Einschränkungen für partitionierte Tabellen • TIMESTAMPDIFF Funktion • TO_DATE Funktion • EBCDIC_STR Funktion • VARCHAR_FORMAT Funktion
IBM DB2 für z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Es wurde ein Fehler behoben, bei dem der funktionsbasierte Index die Funktionen nach der Konvertierung übersprungen hat.
IBM DB2 für z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Es wurde ein Fehler behoben, bei dem die REPEAT Anweisung nach der Konvertierung mit AI 9996 geschlossen wurde

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM DB2 für z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Ein Fehler wurde behoben, bei dem die Klausel mit 9996 geschlossen wurde. FINAL TABLE
IBM DB2 für z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	LOADER Einschränkung für die Partitionierung des Schlüssels in den Referenzen. AWS SCT kann jetzt Primärschlüssel und eindeutige Einschränkungen in partitionierten Tabellen als Sekundärindizes konvertieren.
IBM DB2 für z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Unterstützung der Funktion Postgresql.varchar_format
IBM DB2 für z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Die Kollationsänderung CreateTransformationRule und die ModifyTransformationRule SCT-CLI-Befehle wurden implementiert.
Greenplum	Amazon Redshift	Der Fehler mit dem falschen Aufruf der gespeicherten Prozedur nach der Konvertierung wurde behoben
Hadoop	Amazon EMR	Unterstützung für die Verbindung zu einem Hadoop-Cluster über das rsa-sha2-Protokoll wurde hinzugefügt.
Hadoop	Amazon EMR	Unterstützung für Amazon EMR mit Metastore ohne Glue Hive hinzugefügt,

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Amazon Redshift	Ein Fehler bei der falschen Konvertierung einer rekursiven Abfrage, bei der die Spalte PRIOR nicht in der SELECT-Liste enthalten war, wurde behoben.
Oracle	Aurora PostgreSQL L/ Amazon RDS PostgreSQL	Implementiert, die ein Element eines assoziativen Arrays zurückgibt
Oracle	Aurora PostgreSQL L/ Amazon RDS PostgreSQL	Unerwartetes AI 9996 in Klammern behoben UNPIVOT
Oracle	Aurora PostgreSQL L/ Amazon RDS PostgreSQL	Unerwartetes AI 9996 in mit behoben UNPIVOT UNION ALL
Oracle	Aurora PostgreSQL L/ Amazon RDS PostgreSQL	Verbesserungen für Datentypkonvertierungen Number
Oracle	Amazon Redshift Datenextraktor	Support für die automatische Partitionierung von Oracle-Tabellen. Optimierung für die Erstellung von Migrationsaufgaben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Implementieren Sie die Konvertierung der EXCEPTION BLOCK Aussage
Teradata	Amazon Redshift	Support für die Konvertierung von ALLANY, und SOME Prädikaten nach Amazon Redshift.
Teradata	Amazon Redshift	Native Unterstützung für Prädikate hinzugefügt. QUALIFY
Teradata	Amazon Redshift	Die Konvertierung der folgenden Elemente wurde verbessert: <ul style="list-style-type: none"> • Rekursive Abfragen • GROUPING SET • CUBE • ROLLUP • UPDATEAnweisung mit impliziter Verknüpfung
OLAP-Quellen	Amazon Redshift Datenextraktor	CLI-Befehle für Stop/Resume für Amazon Redshift Data Extractor-Aufgaben implementiert.
OLAP-Quellen	Amazon Redshift Datenextraktor	Es wurde die Möglichkeit hinzugefügt, die Tabellenspalten auszuwählen, die während der Konfiguration der Migrationsaufgabe migriert werden müssen.

Versionshinweise für AWS SCT Build 672

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Amazon RDS für	Unterstützung der PostgreSQL-Hauptversion 15 als Migrationsziel implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
	PostgreSQL	
Alle	Amazon Redshift	Es wurde ein neuer <code>PrintTaskStatus</code> Befehl in der AWS SCT Befehlszeilenschnittstelle (CLI) hinzugefügt, um den Status der Datenmigrationsaufgabe anzuzeigen.
Alle	Amazon Redshift	Der Konfigurationsablauf für die Datenextraktionsagenten wurde verbessert.
Alle	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die Datenextraktionsagenten die Informationen zu Unteraufgaben nicht anzeigten.
Apache Oozie	AWS Step Functions	Es wurde eine Option hinzugefügt, um State-Machine-Definitionen als Skript im konvertierten Code zu speichern.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL	Die Konvertierung von <code>COALESCE</code> , <code>DATEADD</code> / <code>GETDATE</code> , und <code>SUM</code> Funktionen wurde implementiert.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL	Die Konvertierung von <code>UPDATE</code> Aussagen mit <code>JOIN</code> und <code>OUTPUT</code> - Klauseln wurde verbessert.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL	Es wurde ein Fehler behoben, der bei der Konvertierung der <code>SELECT TOP 1 WITH TIES</code> Anweisung aufgetreten ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Es wurden mehrere Probleme behoben, die bei der Konvertierung der FOR XML Klauseln in integrierten Funktionen auftraten.
Greenplum	Amazon Redshift	Die Konvertierung von GET DIAGNOSTICS RAISE AND-Anweisungen wurde mithilfe eines nativen Amazon Redshift EXCEPTION Redshift-Blocks implementiert.
Greenplum	Amazon Redshift	Die Konvertierung von gespeicherten Prozeduren wurde verbessert, indem die Unterstützung eines EXCEPTION Blocks im konvertierten Code hinzugefügt wurde.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Fehler behoben, bei dem TO_CHAR Funktionen mit Zeitformatvorlagen falsch konvertiert wurden.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung verschachtelter Tabellenausdrücke wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung der Anweisungen GOTO, MERGEREPEAT, und SIGNAL wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung von FETCH Anweisungen mit Schlüsselwörtern BEFORE und AFTER Orientierungsschlüsselwörtern wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Die Konvertierung der OLD TABLE Tabellenverweise FINAL TABLE und.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung der folgenden Funktionen wurde implementiert. <ul style="list-style-type: none"> • ADD_MONTHS • DAY mit Parametern des Zeichendatentyps • DAYOFWEEK • DAYS • DECODE • HOUR • LAST_DAY • LOCATE_IN_STRING • MICROSECOND • MINUTE • MONTH • ROUND • TIME • TIMESTAMP • TIMESTAMP_FORMAT • TRANSLATE • UNICODE_STR • XMLCAST • XMLELEMENT • XMLQUERY • XMLSERIALIZE • YEAR
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung eines Alias einer Unterabfrage in Klauseln wurde verbessert. JOIN

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von COALESCE Funktionen wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von EXPLICIT Indizes wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Spaltennamen in zusammengesetzten Ausdrücken wurde verbessert, um ein Problem zu beheben, bei dem das Aktionselement 9997 während der Konvertierung unerwartet angezeigt wurde.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Primärschlüsseln und eindeutigen Einschränkungen wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von XMLTABLE Anweisungen in INSERT Anweisungen wurde verbessert, um ein Problem zu beheben, bei dem während der Konvertierung unerwartet der Aktionspunkt 9996 angezeigt wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet während der Konvertierung von Funktionen mit dem Argument angezeigt wurde. SUBSTR
	PostgreSQL	
IBM Db2 für z/OS	Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet während der Konvertierung des CURRENT_TIMESTAMP Spezialregisters angezeigt wurde.
	PostgreSQL	
IBM Db2 für z/OS	Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet bei der Konvertierung von MERGE Anweisungen, nicht unterstützten Anweisungen und nicht unterstützten integrierten Funktionen angezeigt wurde.
	PostgreSQL	
Microsoft SQL Server	Alle	Unterstützung für Microsoft SQL Server Version 2022 als Quelle hinzugefügt.
Microsoft SQL Server	Aurora PostgreSQL	Die Konvertierung von SELECT Anweisungen, die Operatoren zur Verkettung von Zeichenketten verwenden, wurde verbessert. AWS SCT verwendet die STRING_AGG Funktion im konvertierten Code.
	PostgreSQL	
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Die Unterstützung der neuen Version 3.1.0 der Babelfish-Funktion skonfigurationsdatei wurde implementiert. Diese Datei definiert SQL-Funktionen, die von bestimmten Babelfish-Versionen unterstützt und nicht unterstützt werden.
	L	

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Netezza	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Datenextraktionsagenten die Datenmigration nicht vom angegebenen CDC-Punkt aus starteten.
Oracle	Alle	Der Bewertungsbericht für Oracle-Datenbanken Version 19 wurde als Quelle aktualisiert.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung des DBMS_OUTPUT Pakets wurde implementiert, indem dem AWS SCT Erweiterungspaket neue Funktionen hinzugefügt wurden.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Funktionen und Prozeduren, die assoziative Arrays als Argumente oder Parameter verwenden, wurde implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von DISTINCT Klauseln in Anweisungen wurde verbessert. SELECT
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Tabellen, bei denen die Primärschlüssel einschränkung denselben Namen wie die Tabelle hat, wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung der RAISE_APPLICATION_ERROR Prozedur mit dem dritten Parameter wurde verbessert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem die Migrationsregel den NUMERIC Datentyp, INTEGER sofern zutreffend, nicht automatisch geändert hat.
Oracle DW	Amazon Redshift	Unterstützung für native Amazon Redshift CONNECT BY Redshift-Klauseln im konvertierten Code implementiert.
Oracle DW	Amazon Redshift	Die Datenmigration wurde verbessert, indem automatisch eine Unteraufgabe für jede Tabelle oder Partition im Migrationsbereich hinzugefügt wurde. Dieser Ansatz verhindert den Datenverlust eingefügter Daten nach der Partitionierung.
Teradata	Amazon Redshift	Die Konvertierung von rekursiven Ansichten wurde implementiert.
Teradata	Amazon Redshift	Die Konvertierung von gespeicherten Prozeduren, die ANSI Transaktionsmodi verwendenBTET, wurde verbessert, indem Unterstützung für den nativen Amazon Redshift AUTOCOMMIT Redshift-Transaktionsmodus hinzugefügt wurde.
Teradata	Amazon Redshift	Die Konvertierung von gespeicherten Prozeduren, die TERADATA Transaktionssemantik verwenden, wurde verbessert, indem das NONATOMIC Schlüsselwort dem konvertierten Code hinzugefügt wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem der konvertierte Code die AWS Zugriffsschlüssel-ID und den geheimen Zugriffsschlüssel enthielt.

Versionshinweise für AWS SCT Build 671

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Es wurde ein Fehler behoben, bei dem AWS SCT ich nicht berechtigt war, eine Projektdatei in Windows zu speichern.
Alle	Alle	<p>Die folgenden Vorlagen für die AWS SCT Befehlszeilenschnittstelle (CLI) wurden aktualisiert.</p> <ul style="list-style-type: none"> • BTEQ ScriptConversion • ConversionApply • HadoopMigration • HadoopResumeMigration • Informatica <p>Weitere Informationen zu den AWS SCT CLI-Vorlagen finden Sie unter CLI-Szenarien abrufen.</p>
Alle	Amazon Redshift	Es wurde ein Fehler behoben AWS SCT , bei dem kein Erweiterungspaket in der Befehlszeilenschnittstelle (CLI) erstellt wurde.
Alle	Amazon Redshift	Es wurde ein Problem behoben, bei dem AWS SCT Datenextraktionsagenten die AWS Snowball Konfiguration in der Befehlszeilenschnittstelle (CLI) nicht verwendeten.
Apache Oozie	AWS Step Functions	Unterstützung für die Migration von Apache Oozie zum AWS Step Functions Befehlszeilenschnittstellen-Modus (CLI) implementiert. Nachdem Sie Ihre Hadoop-Workloads zu Amazon EMR migriert haben, können Sie das Workflow-Scheduling-System nun auf das

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		migrieren. AWS Cloud Weitere Informationen finden Sie unter Konvertierung von Apache Oozie zu AWS Step Functions .
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Resolver-Fehler behoben, der für Tabellen und Aliase auftrat.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von INDEX ON Klauseln wurde implementiert.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung der folgenden Objekte wurde verbessert, um unerwartete Aktionselemente zu vermeiden. <ul style="list-style-type: none"> • Batch-Kontoauszüge • Listen von Ausdrücken • Aliase für Tabellen • Temporäre Tabellen • Auslöser • Benutzervariablen
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Analysefehler behoben, der bei Prozeduren aufgetreten ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Fehler behoben, bei dem falsche Namen temporärer Tabellen im konvertierten Code für OBJECT_ID Funktionen AWS SCT verwendet wurden.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurden Probleme behoben, bei denen das Aktionselement 9996 während der Konvertierung der folgenden Codeelemente unerwartet angezeigt wurde. <ul style="list-style-type: none"> • CONVERT-Funktionen • DATEADD-Funktionen • DELETEAnweisungen innerhalb von Inline-Funktionen • IF-Anweisungen. • INSERToder UPDATE Aktionen in einer Spalte • RETURN-Anweisungen. • UPDATEAnweisungen mit komplexen Abfragen oder Funktionen
BigQuery	Amazon Redshift	Unterstützung für BigQuery als Quelle für den Multiserver-Assessment-Prozess hinzugefügt. Weitere Informationen finden Sie unter Einen Bewertungsbericht für mehrere Server erstellen .
Hadoop	Amazon EMR	Die Version des unterstützten Apache Hive JDBC-Treibers, den Sie für die Verbindung zu Ihren Quelldatenbanken verwenden, wurde aktualisiert. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen .
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Der Quellmetadaten-Loader wurde verbessert, um sicherzustellen, dass Quelldatenbankobjekte wie Primärschlüssel, implizite Indizes usw. AWS SCT geladen werden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Resolver-Fehler behoben, der für Spalten in impliziten Cursors auftrat.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Es wurde die Möglichkeit implementiert, die Formatierung von Spaltennamen, Ausdrücken und Klauseln in DML-Anweisungen im konvertierten Code beizubehalten.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von schemaübergreifenden Fremdschlüsseln wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von LENGTH und VARCHAR Funktionen wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von LABEL ON AND-Anweisungen wurde implementiert. DECLARE CONDITION

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von SELECT Anweisungen mit OPTIMIZE FOR Klauseln wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von CREATE TABLE Anweisungen wurde verbessert, indem Standardwerte für alle unterstützten Datentypen hinzugefügt wurden.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von INCREMENT BY Attributen wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von partitionierten Tabellen wurde verbessert, indem die Möglichkeit hinzugefügt wurde, Tabellenpartitionen vom Konvertierungsbereich auszuschließen.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Primärschlüsseldefinitionen mit INCLUDE Spalten wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung der Funktion wurde verbessert. SUBSTRING
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von SET UND-Anweisungen wurde verbessert. DECLARE HANDLER FOR
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung variabler Datentypen wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von XMLTABLE Funktionen wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Der Migrationsablauf wurde verbessert, indem die folgende Reihenfolge implementiert wurde, in der konvertierte Objekte auf die Zieldatenbank angewendet werden: Tabellen, Partitionen, Indizes, Einschränkungen, Fremdschlüssel und Trigger.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet während der Konvertierung von Kommentaren im Quellcode angezeigt wurde.
	PostgreSQL	
IBM Db2 für z/OS	Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9997 unerwartet bei der Konvertierung von Aliasnamen in Klauseln angezeigt wurde. FROM
	PostgreSQL	
IBM Db2 für z/OS	Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9997 unerwartet während der Konvertierung von Cursor-Aliasen angezeigt wurde.
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	Es wurde ein Fehler behoben, bei dem der konvertierte Code unterschiedliche Ergebnisse für SELECT Anweisungen mit ORDER BY Klauseln zurückgab. Da SQL Server und PostgreSQL NULL Werte unterschiedlich behandeln, enthält der konvertierte Code jetzt NULLS FIRST NULLS LAST OR-Klauseln, die sicherstellen, dass Ihr konvertierter Code Ergebnisse in derselben Reihenfolge zurückgibt wie Ihr Quellcode.
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem Datentypen in Tabellenfunktionen falsch konvertiert wurden.
	PostgreSQL	

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
MySQL	Amazon RDS für MySQL	Es wurde ein Problem behoben, bei dem unerwartet einfache Anführungszeichen (' ') in den Namen der Datenbankobjekte im konvertierten Code auftauchten.
Oracle	Aurora PostgreSQL PostgreSQL	Dem Erweiterungspaket wurden neue Ansichten hinzugefügt, um Oracle-Systemansichten zu emulieren, mit denen Sie Informationen über Partitionen und Unterpartitionen anzeigen.
Oracle	Aurora PostgreSQL PostgreSQL	Zwei Funktionen im Erweiterungspaket wurden aktualisiert, um Schemanamen als Argumente im konvertierten Code hinzuzufügen.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Fehler behoben, bei dem nach dem Aktualisieren des Anwendungscodes in der Benutzeroberfläche AWS SCT nicht die richtigen Parameter für die Konvertierung von C++-Anwendungen verwendet wurden.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von CREATE TYPE Anweisungen wurde verbessert, um unerwartete Ausnahmen zu vermeiden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von verschachtelten Tabellen wurde verbessert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Analysefehler behoben, der bei Paketobjekten aufgetreten ist.
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem Objektnamen im konvertierten Code AWS SCT unerwartet gekürzt wurden, wenn die Namenslänge 60 Zeichen überschreitet.
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem Trigger auf Zeilenebene für partitionierte Tabellen falsch konvertiert wurden.
Oracle DW	Amazon Redshift	Unterstützung der automatischen Tabellenpartitionierung für die Datenmigration implementiert. Um die Datenmigration zu beschleunigen, AWS SCT können große Tabellen oder Partitionen automatisch auf der Grundlage der Werte in der ROWID Pseudospalte partitioniert werden. Weitere Informationen finden Sie unter Native Partitionierung verwenden .

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Die Unterstützung nativer MERGE Befehle im konvertierten Amazon Redshift Redshift-Code wurde implementiert. Weitere Informationen zu dem MERGE Befehl in Amazon Redshift finden Sie unter MERGE im Amazon Redshift Database Developer Guide.
Teradata	Amazon Redshift	Die Konvertierung von DELETE UPDATE AND-Anweisungen, die keine expliziten Tabellennamen verwenden, wurde verbessert.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem IN NOT IN Anweisungen falsch konvertiert wurden.

Versionshinweise für AWS SCT Build 670

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure SQL Database	Aurora PostgreSQL	Es wurden Probleme behoben, bei denen das Aktionselement 9996 während der Konvertierung der folgenden Codeelemente unerwartet angezeigt wurde.
Microsoft SQL Server	PostgreSQL	<ul style="list-style-type: none"> • CREATE INDEXAussagen innerhalb INCLUDE von Anweisungen • DECLARE-Anweisungen. • DECLARE . . . TABLE-Anweisungen. • DECLAREmit Standardwerten innerhalb von LOOP Anweisungen • DELETE-Anweisungen. • DROP CONSTRAINT Aussagen innerhalb von ALTER TABLE Aussagen • EXECUTE AS CALLER und REVERT • IIF-Anweisungen. • Listen von Ausdrücken • MONTH()-Funktionen • UPDATE-Anweisungen. • YEAR()-Funktionen

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse Analytics	Amazon Redshift	Unterstützung für Azure Synapse Analytics als Quelle für den Multiserver-Assessment-Prozess hinzugefügt. Weitere Informationen finden Sie unter Einen Bewertungsbericht für mehrere Server erstellen .
Hadoop	Amazon EMR	Unterstützung für die Migration von Hadoop-Clustern zu Amazon EMR im Befehlszeilenschnittstellenmodus (CLI) implementiert. Weitere Informationen finden Sie unter Migration von Big-Data-Frameworks .
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Resolver-Fehler behoben, der für Quelltabellen und -spalten auftrat.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von CASE Ausdrücken wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von CURRENT_DATE Verweisen auf spezielle Register wurde implementiert. Ein Verweis auf ein spezielles Register in Db2 for z/OS ist ein Verweis auf einen Wert, der vom aktuellen Server bereitgestellt wird.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von DATE und POSSTR Funktionen wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Datetime-Konstanten wurde verbessert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Standardwerten für Spalten der folgenden Datentypen wurde verbessert: DATE, TIMESTAMPTZ, und. TIMESTAMP WITH TIME ZONE
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet während der Konvertierung von Anweisungen angezeigt wurde. SELECT INTO
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von DATEDIFF Funktionen wurde verbessert.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Fehler behoben, bei dem ISNULL Funktionen konvertiert wurden NULLIF. Infolgedessen führte der konvertierte Code zu anderen Ergebnissen als der Quellcode. AWS SCT Konvertiert jetzt ISNULL Funktionen in COALESCE.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Netezza	Amazon Redshift	Verbesserte Datenextraktionsagenten zur Behebung eines Problems, bei dem für erfolgreich abgeschlossene Aufgaben der Status „Fehlgeschlagen“ gesetzt wurde.
Netezza	Amazon Redshift	Es wurde die Möglichkeit hinzugefügt, Endpunkte in Unteraufgaben nach dem Start einer Datenmigration mit Datenextraktionsagenten zu ändern.
Microsoft SQL Server MySQL Oracle PostgreSQL	Aurora MySQL Aurora PostgreSQL L MySQL PostgreSQL L	Es wurde die Möglichkeit hinzugefügt, mithilfe eines IPv6-Adressprotokolls eine Verbindung zu Datenbanken herzustellen.
Oracle	Amazon RDS für Oracle	Die Konvertierung des DBMS_JOB Pakets, das Jobs in der Job-Warteschlange plant und verwaltet, wurde implementiert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Dem Erweiterungspaket wurden neue Funktionen hinzugefügt, um die Konvertierung der globalen verschachtelten Tabellen zu verbessern. Diese neuen Funktionen emulieren DELETEEXTEND, und TRIM Funktionen in Ihrem Oracle-Quellcode.
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde die Möglichkeit hinzugefügt, den Konvertierungsbereich für SQL-Code anzugeben, der in Java-Anwendungen eingebettet ist. Sie können jetzt die Teilmengen des Quellanwendungsprojekts vom Konvertierungsumfang ausschließen. Weitere Informationen finden Sie unter Konvertieren Sie den SQL-Code Ihrer Java-Anwendung inAWS SCT .

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Verkettungsoperatoren () innerhalb von Funktionsindizes wurde verbessert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung von IN Bedingungen, bei denen Ihr Quellcode keine Klammern für einen einzelnen Ausdruck enthält.
Oracle	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung von MERGE Anweisungen INSERT ON CONFLICT in PostgreSQL.
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Parsing-Fehler behoben, der bei Prozedurpaketen auftrat.
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem das Aktionselement 5072 während der Konvertierung von Paketen unerwartet angezeigt wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle DW	Amazon Redshift	Es wurde ein Fehler behoben, bei dem das Erweiterungspaket AWS SCT nicht angewendet wurde, wenn der konvertierte Code auf die Zieldatenbank angewendet wurde.
Oracle DW	Amazon Redshift	Es wurde ein Fehler behoben, bei dem einige der Erweiterungspaket-Dateien AWS SCT nicht angewendet wurden, wenn der Erweiterungspaket-Assistent verwendet wurde.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Datenmigration nicht verarbeitet werden AWS SCT konnte, AWS Snowball wenn mehr als 500 Aufgaben parallel ausgeführt wurden.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem benutzerdefinierte Funktionen mit benutzerdefinierten Typen falsch konvertiert wurden.

Versionshinweise für Build 669 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Der Bewertungsprozess für mehrere Server wurde verbessert, mit dessen Hilfe die optimale Zieldatenbankplattform für Ihre Quelldatenbanken ermittelt werden kann. AWS SCT Ignoriert jetzt den AWS Secrets Manager Schlüssel, wenn Sie Datenbankinformationen in der Eingabedatei mit kommagetrennten Werten (CSV) angeben. Weitere Informationen finden Sie unter Einen Bewertungsbericht für mehrere Server erstellen .
Alle	Alle	Es wurde ein Problem behoben, bei dem der Multiserver-Assessmentbericht die IP-Adresse Ihrer Quelldatenbank enthielt, wenn für die Verbindung mit der Datenbank ein geheimes AWS Secrets Manager Formular verwendet wurde.
Alle	Amazon Redshift	Die automatische Konfiguration der Einstellungen für die Java Virtual Machine (JVM) wurde je nach Betriebssystem und verfügbarem RAM

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		implementiert. AWS SCT verwendet diese JVM, um die Arbeit der Datenextraktionsagenten auszuführen.
Alle	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Datenextraktionsagenten in Ubuntu nicht gestartet wurden.
Alle	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Datenextraktionsaufgaben nach dem Ausführen der StartAgent.bat Datei in Windows nicht gestartet wurden.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem Spaltennamen falsch konvertiert wurden, wenn die Option „Eindeutige Namen für Indizes generieren“ aktiviert war.
Greenplum	Amazon Redshift	Die Konvertierung von Funktionen, die VOID zu Prozeduren zurückkehren, wurde implementiert.
Greenplum	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Datenmigration fehlschlug, wenn die Quelldatenbank keine numerischen Werte (NaN) in numerischen Spalten enthielt. AWS SCT Datenextraktionsagenten ersetzen jetzt NaN-Werte durch NULL.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde eine neue Konvertierungseinstellung hinzugefügt, um die TIME FORMAT Optionen DATE FORMAT und während der Konvertierung CHAR integrierter Funktionen festzulegen.
IBM Db2 für z/OS	Aurora PostgreSQL PostgreSQL	Es wurde ein Aktionselement 8534 für die Konvertierung von vordefinierten Cursors hinzugefügt, die mit der Klausel deklariert wurden. WITHOUT RETURN Wenn Ihr Cursor keine Ergebnismengen zurückgibt, AWS SCT weist er Ihrem Cursornamen im konvertierten Code einen NULL Wert zu und löst ein Aktionselement aus.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die CURRENT_CLIENT_APPLNAME Eigenschaft, die AWS SCT während der Verbindung zur Quelldatenbank identifiziert wurde, wurde bearbeitet.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Es wurde eine neue Konvertierungseinstellung implementiert, um die TIME_FORMAT Optionen DATE_FORMAT und bei der Konvertierung CHAR integrierter Funktionen festzulegen.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von LEAVE Anweisungen in BEGIN...END Blockanweisungen wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung vonXMLPARSE, XMLTABLE und XMLNAMESPACES -Funktionen wurde implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung der CHAR integrierten Funktionen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung von Cursors.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet während der Konvertierung von FOR Loop-Anweisungen angezeigt wurde.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung der in SELECT Anweisungen verwendeten Tabellentypen wurde verbessert.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL L	Die Unterstützung der neuen Version 2.2.0 der Babelfish-Funktion skonfigurationsdatei wurde implementiert. Diese Datei definiert SQL-Funktionen, die von bestimmten Babelfish-Versionen unterstützt und nicht unterstützt werden.
Netezza	Amazon Redshift	Verbesserte Datenextraktionsagenten zur Behebung eines Problems, bei dem eine Zeile während der laufenden Datenreplikation nicht aus der Zieltabelle gelöscht wurde.
Oracle	Amazon RDS für Oracle	Verbesserte Konvertierung der Funktionen der Oracle Database Enterprise Edition.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von GROUPING_ID Funktionen wurde implementiert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Die SQL-Codekonvertierung in C#-Anwendungen wurde verbessert, indem Unterstützung für die Zuordnung benutzerdefinierter Datentypen im Befehlszeilenschnittstellenmodus (CLI) hinzugefügt wurde.
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung verschachtelter Tabellen wurde verbessert, um ein unerwartetes Aktionselement 9996 zu vermeiden.
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem der Aufruf eines Objektkonstruktors falsch konvertiert wurde.
Oracle DW	Amazon Redshift	Unterstützung vorhandener Tabellenpartitionen für die Datenmigration implementiert. Um die Datenmigration zu beschleunigen, AWS SCT erstellt Unteraufgaben für jede Partition der Quelltable, die nicht leer ist. Weitere Informationen finden Sie unter Native Partitionierung verwenden .
Teradata	Amazon Redshift	Verbesserte Konvertierung von CAST Funktionen mit TIMESTAMP AS TIME WITH TIME ZONE Argumenten TIME WITH TIME ZONE AS TIMESTAMP TIME WITH TIME ZONE AS CHAR, und.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Verbesserte Konvertierung von CAST Funktionen mit der FORMAT Option.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem CEIL Funktionen nicht konvertiert wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem MERGE Aussagen mit DELETE Klauseln falsch konvertiert wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem TO_CHAR Funktionen mit Datums- und Formatargumenten falsch konvertiert wurden.

Versionshinweise für AWS SCT Build 668

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Amazon Redshift	Es wurde ein Problem behoben, bei dem Multiplikationsoperatoren in Migrationsregeln nicht richtig funktionierten. Diese Operatoren ermöglichen es, die Länge von <code>char</code> , <code>varchar</code> , und <code>string</code> Datentypen zu ändern. Weitere Informationen finden Sie unter Migrationsregeln erstellen .
Azure Synapse Analytics	Amazon Redshift	Unterstützung von CONVERT Funktionen mit VARCHAR Argumenten implementiert.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von SELECT Aussagen mit NOLOCK Klauseln.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von UPDATE Aussagen mit Aliasnamen oder mit SET UND-Klauseln FROM.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Greenplum	Amazon Redshift	Automatische virtuelle Partitionierung für die Datenmigration implementiert. AWS SCT verwendet die GP_SEGMENT_ID Systemspalte, um Partitionen zu erstellen.
Greenplum	Amazon Redshift	Die Unterstützung von RETURN QUERY RETURN SETOF UND-Klauseln wurde implementiert.
Greenplum	Amazon Redshift	Unterstützung von SUBSTRING Funktionen mit drei Parametern implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung von SUBSTR Funktionen mit LOCATE Parametern.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Es wurde eine Option hinzugefügt, um ein Array von REFCURSOR Variablen zur Rückgabe dynamischer Ergebnismengen zu verwenden. Wenn Sie diese Option in den Konvertierungseinstellungen auswählen, wird dem konvertierten Code ein zusätzlicher OUT Parameter AWS SCT hinzugefügt.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Unterstützung von FOR Loop-Anweisungen implementiert.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Unterstützung von XMLPARSE Funktionen implementiert. Es wurde ein Aktionspunkt 8541 für die Markierung von Leerzeichen in XMLPARSE Funktionen hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung mehrerer Ausnahmebehandler in einem einzigen Block wurde verbessert. BEGIN . . . END
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von INSERT und DELETE Triggern wurde verbessert.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung von verschachtelten Prozeduraufrufen.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung von Tabellentypen.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem bitweise logische NOT Operationen fälschlicherweise in Ganzzahlwerte konvertiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem lokale Arrays in PostgreSQL Version 8.0.2 und niedriger nicht initialisiert wurden.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem MERGE Aussagen mit WHEN NOT MATCHED BY SOURCE Klauseln falsch konvertiert wurden.
MySQL	Aurora MySQL	Es wurde ein Problem behoben, bei dem die Benutzerberechtigungen, die von der <code>rds_superuser_role</code> Rolle gewährt wurden, AWS SCT falsch bestimmt wurden.
Netezza	Amazon Redshift	Das Ladeprogramm für Quellmetadaten wurde verbessert, um sicherzustellen, dass Datenbankobjekte mit Namen in Kleinbuchstaben AWS SCT korrekt geladen werden.
Oracle	Aurora PostgreSQL PostgreSQL	Dem Erweiterungspaket wurden neue Funktionen hinzugefügt, um die Konvertierung von lokalen verschachtelten Tabellen zu verbessern. Diese neuen Funktionen emulieren <code>PRIOR</code> , <code>NEXT</code> , <code>LIMIT</code> , <code>FIRST</code> , <code>LAST</code> , <code>EXISTS</code> , <code>EXTEND TRIMDELETE</code> , und <code>SET</code> Funktionen in Ihrem Oracle-Quellcode. Weitere Informationen finden Sie unter Verwenden von Erweiterungspaketen .
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde die Möglichkeit hinzugefügt, den Konvertierungsbereich für C#-Anwendungen anzugeben. Benutzer können jetzt die Teilmengen des Quellenanwendungsprojekts vom Konvertierungsumfang ausschließen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Unterstützung von COUNT Methoden in Sammlungen implementiert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Unterstützung von Variablen und Konstruktoren in verschachtelten Tabellen implementiert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Unterstützung von RATIO_TO_REPORT und STANDARD_HASH Funktionen implementiert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung von großen Objekten (LOBs) als Teil des AWS SCT Erweiterungspakets.
Oracle	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung lokaler Sammlungen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung von JOIN Anweisungen mit USING Klauseln, bei denen Spaltennamen den Tabellennamen nicht enthalten.
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von EMPTY_BLOB und EMPTY_CLOB Funktionen wurde implementiert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Positionsbindungsvariablen in C#-Anwendungen wurde implementiert.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Triggern mit mehreren Ereignissen wurde implementiert.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von rekursiven Triggern wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung von Triggern mit der @@rowcount globalen Variablen.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem Aggregatfunktionen in der SET Klausel von UPDATE Anweisungen falsch konvertiert wurden.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem das Aktionselement 42702 während der Konvertierung von UPDATE Anweisungen unerwartet angezeigt wurde.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem CONVERT Funktionen mit CHAR Argumenten falsch konvertiert wurden.
Snowflake	Amazon Redshift	Unterstützung für Snowflake als Quelle für Datenmigration mit AWS SCT Datenextraktionsagenten hinzugefügt. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift .
Teradata	Amazon Redshift	Verbesserte Konvertierung von CAST Funktionen mit TIMESTAMP AS TIME WITH TIMEZONE Argumenten.

Versionshinweise für AWS SCT Build 667

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Unterstützung von Informatica-Skripts zum Extrahieren, Transformieren und Laden (ETL) im CLI (Command Line Interface) -Modus implementiert. AWS SCT leitet Ihre Informatica ETL-Skripts automatisch zur neuen Zieldatenbank um. AWS SCT Konvertiert außerdem Objektnamen und SQL-Code, der in Ihre Informatica-Objekte eingebettet ist. Weitere Informationen finden Sie unter Konvertierung von Informatica ETL-Skripten .
Alle	Amazon Redshift	Die unterstützte Mindesttreiberversion für Amazon Redshift wurde auf 2.1.0.9 erhöht. Weitere Informationen finden Sie unter Die erforderlichen Datenbanktreiber werden heruntergeladen .
Azure Synapse Analytics	Amazon Redshift	Dem Erweiterungspaket wurde eine neue Funktion hinzugefügt, um die Konvertierung der CONVERT Funktion mit drei Datums- und Uhrzeitargumenten zu verbessern.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung der DATEDIFF Funktion.
Azure Synapse Analytics Microsoft SQL Server DW	Amazon Redshift	Die Version des Erweiterungspakets wurde aktualisiert. Stellen Sie sicher, dass Sie die neueste Version des Erweiterungspakets in Ihren bestehenden AWS SCT Projekten anwenden. Weitere Informationen finden Sie unter Verwenden von Erweiterungspaketen .
BigQuery	Amazon Redshift	Es wurde ein Problem behoben, bei dem die gefilterten Objekte im Befehlszeilenschnittstellenmodus (CLI) nicht konvertiert wurden.
Greenplum	Amazon Redshift	Es wurde ein Fehler behoben, bei dem temporäre Tabellen, die in einer gespeicherten Prozedur deklariert wurden, AWS SCT nicht konvertiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Greenplum	Amazon Redshift	Es wurde ein Fehler behoben, bei dem Spaltencodierungsattribute im konvertierten Code fehlten.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von UPDATE Anweisungen für selbstreferenzierende Tabellen mit mehr als einer INNER JOIN Klausel wurde implementiert.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Unterstützung für deleted temporäre Tabellen, die inserted SQL Server für DML-Trigger verwendet, wurde implementiert.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von benutzerdefinierten Typen in gespeicherten Prozeduren, die in verschiedenen Datenbankschemas erstellt wurden. Es wurde ein Problem behoben, bei dem der Datentyp nicht gefunden werden AWS SCT konnte und der Aktionspunkt 9996 angezeigt wurde.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem im konvertierten Code unerwartet eckige Klammern ([]) um die Namen der Datenbankobjekte standen.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem @@ROWCOUNT Funktionen falsch konvertiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server DW	Amazon Redshift	Unterstützung von geometry geography Datentypen wurde implementiert.
Microsoft SQL Server DW	Amazon Redshift	Die Unterstützung des MAX Schlüsselworts in Datentypdeklarationen im konvertierten Code wurde implementiert.
Microsoft SQL Server DW	Amazon Redshift	Verbesserte Konvertierung von DATEADD Funktionen.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte SQL-Code-Konvertierung in Java-Anwendungen durch Hinzufügen von Unterstützung für das MyBatis Framework. Weitere Informationen finden Sie unter Umwandeln von SQL-Code in Java-Anwendungen .
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte SQL-Codekonvertierung in Java-Anwendungen, die das MyBatis Framework verwenden. Es wurde ein Aktionselement 30411 für SQL-Code mit nicht unterstützter Syntax hinzugefügt.
Oracle	Aurora PostgreSQL PostgreSQL	Die SQL-Codekonvertierung in Pro*C-Anwendungen wurde verbessert, indem Unterstützung für Deklarationen hinzugefügt wurde. typedef struct

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Unterstützung von UND-Anweisungen CROSS JOIN implementiert. LEFT JOIN
Oracle	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung von MERGE Aussagen. Es wurde ein Problem behoben, bei dem einzufügende Werte im konvertierten Code fehlten.
Teradata	Amazon Redshift	Die standardmäßigen Codierungseinstellungen für die Spaltenkomprimierung, die im konvertierten Code AWS SCT verwendet werden, wurden so geändert, dass sie den Standardeinstellungen von Amazon Redshift entsprechen. Weitere Informationen finden Sie unter Kompressionskodierungen im Amazon Redshift Database Developer Guide.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem mathematische Operationen, die den TIME Datentyp verwenden, falsch konvertiert wurden.
Teradata	Amazon-Redshift-RSQL	Die Konvertierung von FastExport Code, der sich in Shell-Skripten befindet, wurde implementiert.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Fehler behoben, bei dem Anweisungen AWS SCT nicht konvertiert wurden. COALESCE %data
Vertica	Amazon Redshift	Verbesserte Vorschläge zur Konversionsoptimierung, wenn ein Benutzer eine Optimierungsstrategie auswählt.

Versionshinweise für AWS SCT Build 666

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure SQL Database	Aurora PostgreSQL	Es wurde ein Analysefehler behoben, der bei ON Klauseln auftrat, die sich in JOIN Anweisungen befinden.
Microsoft SQL Server	PostgreSQL	
Azure Synapse Analytics	Amazon Redshift	Dem Erweiterungspaket wurden drei neue Funktionen hinzugefügt, um die Konvertierung der CONVERT Funktion mit Datums- und Uhrzeitargumenten zu verbessern.
Azure Synapse Analytics	Amazon Redshift	Der Quellmetadaten-Loader wurde verbessert, um sicherzustellen, dass Systemdatenbankschemas AWS SCT geladen werden.
Azure Synapse Analytics	Amazon Redshift	Es wurde ein Resolver-Fehler behoben, der bei Spalten temporärer Tabellen auftrat.
Azure Synapse Analytics	Amazon Redshift	Die Konvertierung von BINARY VARBINARY Datentypen in den VARBYTE Datentyp wurde implementiert.
Azure Synapse Analytics	Amazon Redshift	Die Unterstützung des TIME Datentyps im konvertierten Code wurde implementiert.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von COLLATE Klauseln. Es wurde ein Problem behoben, bei dem das Aktionselement 31141 bei der Konvertierung von Spalten mit der Standardsortierung der Datenbank unerwartet angezeigt wurde.
BigQuery	Amazon Redshift	Die Konvertierung von Prozeduren, die Eingabeparameter ändern, wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Greenplum	Amazon Redshift	Es wurde ein Problem behoben, bei dem eine Abfrage AWS SCT verwendet wurde, die nicht mit Greenplum 6.x-Datenbanken kompatibel ist.
IBM Db2 für z/OS	Aurora PostgreSQL	Verbesserte Konvertierung von EXCEPTION Abschnitten durch Übertragung von Ausnahmehandlern von Db2 for z/OS nach PostgreSQL.
	PostgreSQL	
IBM Db2 für z/OS	Aurora PostgreSQL	Verbesserte Konvertierung von Kontoauszügen. OPEN CURSOR
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	Die Konvertierung von IIF Funktionen mithilfe von CASE Ausdrücken wurde implementiert.
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem Prozeduren mit Tabellenwertparametern falsch konvertiert wurden, wenn die CREATE PROCEDURE Anweisung keinen BEGIN . . . END Block enthielt.
	PostgreSQL	

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem die SCOPE_IDENTITY Funktion falsch konvertiert wurde.
Oracle	Amazon RDS für Oracle	Es wurde ein Loader-Fehler behoben, der bei der SELECT_CATALOG_ROLE Rolle auftrat, wenn Oracle 10g als Quelle verwendet wurde.
Oracle	Amazon RDS für Oracle	Der Loader wurde verbessert, sodass er Oracle Scheduler-Jobs unterstützt.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung von JOIN Anweisungen mit USING Klauseln wurde implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Leistung des konvertierten Codes, wenn der Quellcode globale Variablen in WHERE Klauseln enthält.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte SQL-Code-Konvertierung in Java-Anwendungen durch Hinzufügen von Unterstützung für das MyBatis Framework. Weitere Informationen finden Sie unter Umwandeln von SQL-Code in Java-Anwendungen .

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle DW	Amazon Redshift	Die Konvertierung der Operatoren PIVOT und der UNPIVOT relationalen Operatoren wurde implementiert.
Teradata	Amazon Redshift	Es wurde ein Fehler behoben, bei dem Quellcode, der JSON-Objekte verwendet, nicht konvertiert wurde.
Teradata	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die von einem gelöschten Benutzer erstellten Tabellen nicht korrekt geladen wurden.
Teradata	Amazon Redshift	Die Konvertierung von INSTR Funktionen in native Amazon Redshift STRPOS Redshift-Funktionen wurde implementiert.
Teradata	Amazon Redshift	Die Konvertierung der TRANSLATE Funktionen NVP und wurde implementiert.
Teradata	Amazon Redshift	Verbesserte Konvertierung von COALESCE Ausdrücken.
Teradata	Amazon Redshift	Verbesserte Konvertierung von DECLARE CONDITION Kontoauszügen.
Teradata	Amazon Redshift	Verbesserte Konvertierung von EXTRACT Funktionen mit dem SECOND Syntaxelement.
Teradata	Amazon Redshift	Verbesserte Konvertierung der SQLCODE Variablen SQLSTATE und in den LOOP Anweisungen.
Teradata	Amazon Redshift	Verbesserte Konvertierung von eindeutigen Indizes.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 unerwartet bei der Konvertierung von CURRENT_TIMESTAMP Anweisungen angezeigt wurde, deren Genauigkeit auf 3 gesetzt war.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem Backslashes fälschlicherweise in Zeichenfolgenliterals umgewandelt wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem konvertierte EXEC Anweisungen einen falschen Feldnamen in der Anweisung enthielten. ADD CONSTRAINT
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem konvertierte QUALIFY Unterabfragen einen falschen Unterabfragenamen enthielten.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem konvertierte Ansichten nicht angewendet wurden. Es wurde eine explizite Umwandlung in einen bestimmten Datentyp für NULL-Werte im konvertierten Code hinzugefügt.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem Datums- und Uhrzeitfunktionen falsch konvertiert wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem hexadezimale Zeichenkettenlitterale nicht konvertiert wurden.

Versionshinweise für Build 665 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse Analytics	Amazon Redshift	Die Konvertierung von CONCAT Funktionen mit VARCHAR Argumenten wurde implementiert.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von CREATE TABLE Anweisungen, die temporäre Tabellen erstellen und den Schemanamen nicht enthalten. AWS SCT erstellt das dbo Schema zum Speichern dieser temporären Tabellen in der Zieldatenbank.
Azure Synapse Analytics	Amazon Redshift	Die Konvertierung von DROP TABLE Anweisungen, die Sie in temporären Tabellen ausführen, wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von OBJECT_ID Anweisungen mit den BEGIN...END Blöcken.
Azure Synapse Analytics	Amazon Redshift	Es wurde ein Fehler behoben, bei dem gespeicherte Prozeduren mit Blockkommentaren nicht konvertiert werden AWS SCT konnten.
BigQuery	Amazon Redshift	Die Umstellung von BigQuery Data Warehouses auf Amazon Redshift wurde implementiert. Weitere Informationen finden Sie unter Verwenden von BigQuery als Quelle für AWS SCT .
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von Triggern, die mehrere Ereignisse verarbeiten inserted und mit deleted Systemtabellen in SQL Server arbeiten.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Resolver-Fehler behoben, der für inserted deleted Systemtabellen in SQL Server auftrat.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Die Unterstützung der neuen Version 2.1.0 der Babelfish-Funktion skonfigurationsdatei wurde implementiert. Diese Datei definiert SQL-Funktionen, die von bestimmten Babelfish-Versionen unterstützt und nicht unterstützt werden.
Oracle	Aurora MySQL MariaDB MySQL	Es wurde ein Problem behoben, bei dem der varchar2 Datentyp falsch konvertiert wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora MySQL	AWS SCT Unterstützt für Oracle-Datenbanken ab Version 12c die folgenden erweiterten Datentypen:
	Aurora PostgreSQL L	<ul style="list-style-type: none"> • VARCHAR2 • NVARCHAR2 • RAW
	MariaDB MySQL PostgreSQL L	AWS SCT Die maximal unterstützte Spaltenlänge für diese Datentypen wurde von 8.000 auf 32.767 Byte erhöht.
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Analysefehler behoben, der für das Oracle Event Processing Package aufgetreten ist.
Teradata	Amazon Redshift	Es wurde ein Aktionspunkt 13214 für mehrere RESET WHEN Klauseln in einer einzigen Anweisung hinzugefügt. SELECT
Teradata	Amazon Redshift	Es wurde ein Aktionselement für SQLSTATE Variablen hinzugefügt, die sich außerhalb eines Ausnahmebehandlungsblocks befinden.
Teradata	Amazon Redshift	Die Konvertierung von ACTIVITY_COUNT Variablen in wurde implementiert ROW_COUNT .
Teradata	Amazon Redshift	Die Konvertierung der integrierten ST_TRANSFORM Geometrie funktion wurde implementiert.
Teradata	Amazon Redshift	Verbesserte Konvertierung von Löschanweisungen in Ansichten ohne die WHERE Klausel.
Teradata	Amazon Redshift	Verbesserte Konvertierung von CAST Operatoren in Ausdrücken.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Verbesserte Konvertierung von GROUP BY Klauseln.
Teradata	Amazon Redshift	Verbesserte Konvertierung von INSTR und REGEXP_INSTR integrierten Funktionen.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem Aliasverweise auf seitliche Spalten falsch konvertiert wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem Spaltennamen in der QUALIFY Unterabfrage falsch konvertiert wurden.
Teradata	Amazon Redshift	Die Konvertierung von .QUIT Befehlen mit dem Schlüsselwort ERRORCODE status value wurde implementiert.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9996 während der Konvertierung von Anweisungen unerwartet angezeigt wurde. CREATE
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9998 während der Konvertierung von Anweisungen unerwartet angezeigt wurde. END

Versionshinweise für AWS SCT Build 664

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Unterstützung für Amazon Redshift Serverless als Quelle und Ziel für Datenbankmigrationsprojekte in hinzugefügt. AWS SCT Um eine Verbindung zu Amazon Redshift Serverless herzustellen, stellen Sie sicher, dass Sie den Amazon Redshift JDBC-Treiber Version 2.1.0.9 oder höher verwenden.
Alle	Alle	Die Benutzeroberfläche des Fensters mit den Konvertierungseinstellungen wurde verbessert. AWS SCT zeigt jetzt nur Einstellungen

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen	
		für Datenbankkonvertierungspaare mit erstellten Zuordnungsregeln an. Weitere Informationen finden Sie unter Mapping-Regeln erstellen .	
Alle	Alle	Der Bewertungsbericht wurde aktualisiert, sodass doppelte Informationen über die Linie und Position des Aktionspunkts entfernt wurden.	
Alle	Amazon Redshift	Automatischer Speicherausgleich bei Datenextraktionsaufgaben implementiert.	
Alle	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die Datenextraktionsagenten keine Verbindung zu AWS Snowball Geräten herstellen konnten.	
Azure SQL Database	Aurora MySQL	Die Unterstützung von SUSE Linux 15.3 als Plattform für die Ausführung von Datenextraktionsagenten wurde implementiert.	
IBM Db2 für z/OS	Aurora PostgreSQL		
IBM Db2 (LUW)	MariaDB		
Microsoft SQL Server	MySQL		
MySQL	PostgreSQL		
Oracle	L		
PostgreSQL	L		
SAP ASE	L		
Azure Synapse Analytics	Amazon Redshift		Verbesserte Konvertierung von DATEADD Funktionen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Es wurde die Möglichkeit hinzugefügt, die Spaltensortierung in Migrationsregeln zu ändern.
Microsoft SSIS	AWS Glue AWS Glue Studio	Es wurde ein unerwarteter Fehler behoben, der auftrat, wenn Benutzer ein Quellskript auswählten.
Oracle	Aurora MySQL MariaDB MySQL	Die Konvertierung der Verwendung von gespeicherten Funktionen als generierte Spaltenausdrücke wurde implementiert. AWS SCT erzeugt Trigger, um dieses Verhalten zu emulieren, da MySQL die Verwendung von gespeicherten Funktionen als generierte Spaltenausdrücke nicht unterstützt.
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Funktionen aus dem UTL_MATCH Paket wurde als Teil des AWS SCT Erweiterungspakets implementiert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung der REGEXP_LIKE Funktion mit dem NULL Parameter wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung der SYS_EXTRACT_UTC Funktion.
Oracle	Aurora PostgreSQL L PostgreSQL L	Verbesserte SQL-Codekonvertierung in C++-Anwendungen durch Implementierung der Unterstützung von WcscatsWcscpys, und Wcsncats Funktionen. Weitere Informationen finden Sie unter Konvertieren von SQL-Code in C++-Anwendungen mitAWS SCT .
Oracle DW Snowflake	Amazon Redshift	Es wurde ein Problem behoben, bei dem konvertierte Anweisungen keine explizite Konvertierung von Werten in den Spaltendatentyp enthielten. Dieses Problem trat bei Anweisungen auf, die Abfrageergebnisse aus anderen Tabellen verwenden.
Teradata	Amazon Redshift	Es wurde die Möglichkeit hinzugefügt, die Spaltensortierung zwischen case sensitive und case insensitive in Migrationsregeln zu ändern. Weitere Informationen finden Sie unter Migrationsregeln erstellen .
Teradata	Amazon Redshift	Es wurde ein Resolver-Fehler behoben, der bei Anweisungen auftrat. CREATE TABLE AS
Teradata	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die integrierte P_INTERSECT Funktion mit einem COALESCE Ausdruck nicht konvertiert wurde.
Teradata	Amazon Redshift	Die Konvertierung von Spalten mit dem Namen OID nach wurde implementiert OID, um die Verwendung eines reservierten Schlüsselworts in Amazon Redshift zu vermeiden.
Teradata	Amazon Redshift	Die Konvertierung von RENAME Anweisungen für Funktionen, Prozeduren, Ansichten und Makros wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Die Konvertierung der STROKE Funktion zur SPLIT_PART Funktion in Amazon Redshift wurde implementiert.
Teradata	Amazon Redshift	Verbesserte Konvertierung der Funktionen INSTR und des REGEXP_INSTR Systems.
Teradata	Amazon Redshift	Verbesserte Konvertierung des TIME Datentyps.
Teradata	Amazon Redshift	Verbesserte Emulation der SET MULTISET UND-Tabellen durch Implementierung der Konvertierung primärer und sekundärer eindeutiger Indizes.
Teradata	Amazon Redshift	Es wurde ein Analysefehler behoben, der bei der Funktion aufgetreten ist. CHARACTER
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Fehler behoben, der auftrat, als Benutzer Teradata Basic Teradata Query-Skripts (BTEQ) aus dem Projekt entfernten. AWS SCT

AWS SCT Versionshinweise für Build 663

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Es wurde die Möglichkeit hinzugefügt, die Länge von <code>char</code> , <code>varchar</code> , und <code>string</code> Datentypen mithilfe des Multiplikationsoperators in einer Migrationsregel zu ändern. Weitere Informationen finden Sie unter Migrationsregeln erstellen .
Alle	Alle	Die Unterstützung von drei neuen Spalten im Multiserver-Assessment-Bericht wurde implementiert und das Format der Eingabedatei aktualisiert. Stellen Sie sicher, dass Sie die aktualisierte Vorlage der Eingabedatei mit der neuesten Version von AWS SCT verwenden. Weitere Informationen finden Sie unter Erstellung eines Multiserver-Bewertungsberichts für die Datenbankmigration .

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung von OBJECT_ID Kontoauszügen.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Unterstützung für Babelfish for Aurora PostgreSQL 1.2.0 als Zielplattform für Bewertungsberichte zur Datenbankmigration hinzugefügt. Weitere Informationen finden Sie unter Unterstützte Funktionen in Babelfish nach Version im Amazon Aurora Aurora-Benutzerhandbuch.
Microsoft SQL Server DW	Amazon Redshift	Unterstützung für AT TIME ZONE Klauseln hinzugefügt.
Microsoft SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem eine Anweisung außerhalb des BEGIN/END Blocks falsch konvertiert wurde.
Netezza	Amazon Redshift	Die Konvertierung des TIME Datentyps wurde verbessert und die Konvertierung verwandter integrierter Funktionen, Ausdrücke und Literale implementiert.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Loader-Fehler behoben, der auftrat, wenn Oracle 10g als Quelle verwendet wurde.
Oracle	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von OFFSET FETCH UND-Klauseln.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem Prozeduren mit OUT Parametern mit Standardwerten falsch konvertiert wurden.
Oracle DW	Amazon Redshift	Verbesserte Konvertierung von Oracle-Funktionen in benutzerdefinierte Funktionen von Amazon Redshift.
Snowflake	Amazon Redshift	Verbesserte Konvertierung von Klauseln. WITH
Teradata	Amazon Redshift	Es wurde ein neues Aktionselement 13209 für nicht unterstützte Multibyte-Zeichen für den Datentyp hinzugefügt. CHAR
Teradata	Amazon Redshift	Es wurde ein Loader-Fehler behoben, bei dem die Tabellen nicht vollständig geladen wurden.
Teradata	Amazon Redshift	Es wurde ein Transformer-Fehler behoben, bei dem die eingebaute P_INTERSECT Funktion in einer JOIN Bedingung nicht konvertiert wurde.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem der Name einer Ansicht falsch konvertiert wurde, wenn die SELECT Anweisung für eine Tabelle mit Sonderzeichen im Namen ausgeführt wurde.
Teradata	Amazon Redshift	Die Konvertierung von INSERT Anweisungen mit dem UNTIL_CHANGED Wert im PERIOD(DATE) Datentyp wurde verbessert.
Teradata	Amazon Redshift	Verbesserte Konvertierung der integrierten FORMAT Funktion mithilfe der TO_CHAR Funktion in Amazon Redshift.
Teradata	Amazon Redshift	Die Konvertierung der integrierten RANK Funktion wurde verbessert, um sicherzustellen, dass der konvertierte Code NULL-Werte in derselben Reihenfolge wie der Quellcode zurückgibt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Verbesserte Konvertierung von eindeutigen Einschränkungen wie primären oder sekundären eindeutigen Indizes.

Versionshinweise für AWS SCT Build 662

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Es wurde die Möglichkeit hinzugefügt, bei der Erstellung des Multiserver-Assessment-Berichts automatisch AWS SCT Projekte für jede Quelldatenbank zu erstellen. Wenn diese Option aktiviert ist, AWS SCT können Sie diesen Projekten Zuordnungsregeln hinzufügen und Konvertierungsstatistiken für die Offline-Verwendung speichern. Weitere Informationen finden Sie unter Erstellung eines Multiserver-Bewertungsberichts für die Datenbankmigration .
Alle	Alle	Die Unterstützung von Prozent (%) als Platzhalter in Datenbank- und Schemanamen bei der Erstellung des Multiserver-Assessment-Berichts wurde implementiert.
Alle	Aurora MySQL Aurora PostgreSQL	Die Laufzeit aller AWS Lambda Funktionen wurde auf Python-Version 3.9 aktualisiert.
Alle	Amazon Redshift	Alle Datenextraktionsagenten wurden aktualisiert, sodass sie jetzt verwendet AWS SDK for Java 2.x werden können.
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von DELETE Aussagen mit NON EXISTS Klauseln wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse Analytics	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die Verbindung zu einer Quelldatenbank fehlschlug.
IBM Db2 für z/OS	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Fehler behoben, bei dem der konvertierte Code eines Triggers zwei Erwähnungen des Objektalias enthielt.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Objekten mit Namen in gemischter Groß- und Kleinschreibung wurde verbessert, wenn die Option Datenbank objektnamen mit Groß- und Kleinschreibung behandeln aktiviert ist.
Microsoft SQL Server DW Teradata	Amazon Redshift	Die Konvertierung der Operatoren PIVOT und UNPIVOT relationaler Operatoren wurde implementiert.
Netezza	Amazon Redshift	Die Konvertierung des TIME Datentyps wurde implementiert.
Oracle	Aurora MySQL Aurora PostgreSQL L MySQL PostgreSQL L	Die konstante Konvertierung des UTL_TCP.CRLF Pakets wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem mit Erweiterungspaketen behoben, bei dem die Länge von Datentypen für Spalten mit variabler Länge während der Konvertierung nicht beibehalten wurde.
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von SQL-Code in C++-Anwendungen wurde implementiert. Weitere Informationen finden Sie unter Konvertieren von SQL-Code in C++-Anwendungen mitAWS SCT .
Oracle	Aurora PostgreSQL L PostgreSQL L	Unterstützung für die Benennung von Groß- und Kleinschreibung für die Konvertierung von globalen Variablen und assoziativen Arrays implementiert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung der TO_NUMBER Funktionen TO_CHAR TO_DATE, und im Erweiterungspaket wurde verbessert.
Oracle	Aurora PostgreSQL L PostgreSQL L	Verbesserte Konvertierung des TABLE () Operators.
Oracle DW	Amazon Redshift	Unterstützung für die Konvertierung von Primärschlüsseln und andere Einschränkungen hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem das Aktionselement 12054 bei der Konvertierung von bedingten Anweisungen nicht angezeigt wurde.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Fehler behoben, bei dem während der Konvertierung von Tabellen mit Spalten benutzerdefinierten Typs ein Objekt mit einem leeren Namen im Zielbaum erstellt wurde.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Ein Loader-Fehler für gespeicherte Objekte wie Skripten, Routinen usw. wurde behoben.
Snowflake	Amazon Redshift	Es wurde ein Problem behoben, bei dem der Aktionspunkt 22152 bei Bedarf nicht angezeigt wurde und das Konvertierungsergebnis als Kommentar AWS SCT angezeigt wurde.
Snowflake	Amazon Redshift	Verbesserte Konvertierung der Datums- und Uhrzeitfunktionen; Unterstützung von Zeitzonen implementiert.
Snowflake	Amazon Redshift	Es wurde ein Problem behoben, bei dem nicht-rekursive Common Table Expressions (CTEs) mit einer WITH Klausel in rekursive CTEs konvertiert wurden.
Teradata	Amazon Redshift	Die Konvertierung von UPDATE Anweisungen mit Tabellenverknüpfungen in einer bestimmten Bedingung wurde verbessert.
Teradata	Amazon Redshift	Verbesserte Konvertierung von RENAME TABLE Kontoauszügen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem leere Spalten in der Datei mit kommagetrennten Werten (CSV) mit einem Bewertungsbericht auftauchten.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Fehler behoben, bei dem am Ende des konvertierten BTEQ-Makros (Basic Teradata Query) ein Semikolon fehlte.
Teradata	Amazon-Redshift-RSQL	Die Konvertierung mehrerer Datentypwerte in Anweisungen wurde verbessert. CASE
Teradata	Amazon-Redshift-RSQL	Die Konvertierung der LIKE ANY Klausel mit einem ESCAPE Zeichen wurde verbessert.
Teradata	Amazon-Redshift-RSQL	Verbesserte Konvertierung der CAST Funktion in INSERT Anweisungen.
Teradata	Amazon-Redshift-RSQL	Verbesserte Konvertierung der Zeitzonen, Implementierung der Zuordnung von Zeitzonenregionen.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem das Aktionselement 9998 unerwartet während der Konvertierung von Shell-Skripten mit BTEQ-Skripten angezeigt wurde.
Teradata	Amazon-Redshift-RSQL AWS Glue	Das Limit von 500 Zeichen für die Werte von Substitutionsvariablen wurde implementiert.
Vertica	Amazon Redshift	Die Konvertierung der RAW Datentypen BINARYVARBINARY ,LONG BINARY,BYTEA, und in den VARBYTE Datentyp wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Vertica	Amazon Redshift	Verbesserte Konvertierung der integrierten Funktionen und Literale.

Versionshinweise für AWS SCT Build 661

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Es wurden Filter hinzugefügt, um in der Mapping-Ansicht nach Zuordnungsregeln zu suchen. Wenn Sie einen Filter anwenden, AWS SCT werden Regeln angezeigt, die den Filterbedingungen in der Liste der Serverzuordnungen entsprechen. Weitere Informationen finden Sie unter Verwaltung von Zuordnungsregeln .
Alle	Alle	Apache Log4j wurde auf Version 2.17.1 aktualisiert.
Alle	Amazon Redshift	Unterstützung für die Datenmigration zu Amazon Redshift mithilfe der ENCRYPTED Klausel im COPY Befehl hinzugefügt.
Alle	Amazon Redshift	Die REST-API der Datenextraktionsagenten wurde verbessert. Die aktualisierte REST-API bietet Unterstützung für neue Eigenschaften wie Verschlüsselungsschlüssel, Verschlüsselungstyp usw.
Alle	Amazon Redshift	Die Übernahme einer Rolle in den Datenextraktionsagenten wurde implementiert. Dieses Update verbessert die Verteilung der Unteraufgaben und ermöglicht AWS SCT die Zuweisung von Aufgaben an freie Agenten mit der angegebenen Rolle.
Alle	Amazon Redshift	Es wurde eine Überprüfung implementiert, ob alle erforderlichen Komponenten installiert sind, bevor das Erweiterungspaket auf Amazon Redshift angewendet wird.
Azure Synapse Analytics	Amazon Redshift	Verbesserte Konvertierung der ERROR_STATE Systemfunktionen ERROR_LINE ERROR_MESSAGE ERROR_NUMBER ,ERROR_PROCEDURE ,ERROR_SEVERITY ,, und zur Fehlerbehandlung.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server DW		
IBM Db2 für z/OS	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Unterstützung für IBM Db2 for z/OS Version 12 als Quelle für Datenbankmigrationsprojekte in hinzugefügt. AWS SCT Weitere Informationen finden Sie unter IBM Db2 for z/OS als Quelle verwenden .
IBM Db2 (LUW)	Alle	Der Quellmetadaten-Loader wurde verbessert, um sicherzustellen, dass Routineparameter AWS SCT geladen werden, die Spaltennamen duplizieren.
Microsoft Azure SQL Database	Aurora PostgreSQL	Ein Transformer-Fehler für Prozeduren mit der SET NOCOUNT ON Set-Anweisung wurde behoben.
Microsoft SQL Server	PostgreSQL	
Microsoft Azure SQL Database	Aurora PostgreSQL	Verbesserte Konvertierung der CONCAT Funktion, wenn ein Eingabewert eine Variable vom benutzerdefinierten Typ ist.
Microsoft SQL Server	PostgreSQL	

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft Azure SQL Database	Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem die DATEPART Funktion falsch konvertiert wurde.
Microsoft SQL Server	PostgreSQL	
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Die Unterstützung der neuen Version der Babelfish-Funktionskonfigurationsdatei wurde implementiert. Diese Datei definiert SQL-Funktionen, die von bestimmten Babelfish-Versionen unterstützt und nicht unterstützt werden.
Microsoft SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem Prozeduren mit einer EXECUTE Anweisung falsch konvertiert wurden.
Microsoft SSIS	AWS Glue	Die Benutzeroberfläche des Assistenten zur Jobkonfiguration wurde verbessert. AWS SCT zeigt jetzt nur verfügbare Verbindungen im Abschnitt zur Verbindungskonfiguration an.
Microsoft SSIS	AWS Glue	Es wurde ein Problem behoben, bei dem die Transformationsregeln nicht auf Paketaufgaben und Variablenregeln angewendet wurden.
Microsoft SSIS	AWS Glue AWS Glue Studio	Es wurde ein neuer Aktionspunkt 25042 für nicht unterstützte Komponenten hinzugefügt.
Microsoft SSIS	AWS Glue Studio	Die Konvertierung von Microsoft SQL Server Integration Services (SSIS) -Paketen zum Extrahieren, Transformieren und Laden (ETL) in wurde implementiert. AWS Glue Studio Weitere Informationen finden Sie unter SSIS konvertieren in AWS Glue Studio .
Oracle	MariaDB	Ein Problem mit der Konvertierung des MINUS Operators wurde behoben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	MariaDB	Verbesserte Konvertierung der ADD_MONTHS Funktion nROWNUM, SYS_GUID, und TO_CHAR, wenn die sql_mode Systemvariable in MariaDB auf Oracle gesetzt ist.
Oracle	PostgreSQL	Es wurde eine Option hinzugefügt, um die Konvertierung von Bind-Variablentypen in SQL-Typen in Konvertierungsprojekten für generische Anwendungen zu vermeiden.
Oracle	PostgreSQL	Es wurde eine Option hinzugefügt, um zu verhindern, dass der Schemaname dem Namen des konvertierten Objekts in Konvertierungsprojekten für generische Anwendungen hinzugefügt wird.
Oracle	PostgreSQL	Unterstützung für das ?x Bind-Variablenformat für die Konvertierung von Anwendungs-SQL-Code hinzugefügt.
Oracle DW	Amazon Redshift	Die Konvertierung des RAW Datentyps in den VARBYTE Datentyp wurde implementiert.
Teradata	Amazon Redshift	Es wurde eine Option hinzugefügt, um SET Tabellen im konvertierten Code zu emulieren. Für diese Emulation gibt es AWS SCT Unterstützungen MIN und MAX Bedingungen.
Teradata	Amazon Redshift	Die Konvertierung von Join-Operationen mit Parametern verschiedener Datentypen wurde verbessert. Dieses Update ermöglicht AWS SCT die Anwendung von Transformationsregeln bei der Konvertierung solcher Operationen.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die GROUP BY Klausel falsch konvertiert wurde.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die QUALIFY Klausel falsch konvertiert wurde.
Teradata	Amazon Redshift	Es wurde ein unerwarteter Fehler behoben, der beim Import von FastExport Skripten aufgetreten ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Es wurde die Möglichkeit implementiert, die Werte von Variablen in Teradata BTEQ- und Shell-Skripten zu bearbeiten.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem das Manifest-Skript für die konvertierten Teradata-Sitzungen fehlte. FastLoad
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem die Erweiterung der Manifestdatei im Uniform Resource Locator (URL) für die konvertierten Skripts fehlte. FastLoad
Teradata BTEQ	Amazon-Redshift-RSQL	Ein Loader-Fehler für Skripte mit Substitutionsvariablen wurde behoben.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem Aktionselement 27022 nicht angezeigt wurde, wenn es erforderlich war.

Versionshinweise für AWS SCT Build 660

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Unterstützung für AWS Secrets Manager und Secure Sockets Layer (SSL) wurde im Multiserver-Assessment-Bericht hinzugefügt. Weitere Informationen finden Sie unter Erstellung eines Multiserver-Bewertungsberichts für die Datenbankmigration .
Alle	Alle	Die Erfassung von Statistiken für konvertierte Objekte wurde verbessert.
Alle	PostgreSQL	Implementierte Unterstützung von PostgreSQL-Hauptversion 14 und MariaDB 10.6 als Migrationsziele.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure Synapse Analytics	Amazon Redshift	Verbesserte Transformationslogik für die Namen konvertierter Objekte.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL	Verbesserte Konvertierung des XML Datentyps.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem NOT LIKE Klauseln falsch konvertiert wurden.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL	Ein Transformationsfehler für Prozeduren mit INSERT, und UPDATE -Anweisungen DELETE, die die OUTPUT Klausel enthalten, wurde behoben.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL	Ein Transformationsfehler für Prozeduren mit der RETURN @@ROWCOUNT Anweisung wurde behoben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Alle	Die Konvertierung von Prozeduren, die Verbindungsserver verwenden, wurde verbessert.
Microsoft SQL Server	Alle	Unterstützung der Microsoft Windows-Authentifizierung im Multiserver-Assessment-Bericht hinzugefügt.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Ein Transformationsfehler für Tabellenwertkonstruktoren wurde behoben.
Microsoft SQL Server DW	Amazon Redshift und AWS Glue	Die Konvertierung von ETL-Skripten (Extrahieren, Transformieren und Laden) wurde verbessert, sodass nun der richtige Pfad zu den konvertierten Skripten enthalten ist.
Microsoft SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem verschiedene konvertierte Skripts für virtuelle und reale Zieldatenbankplattformen generiert wurden.
Oracle	PostgreSQL Aurora PostgreSQL	Unterstützung für die Konvertierung von Indizes für materialisierte Ansichten hinzugefügt.
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem der Aktionspunkt 5982 bei der Konvertierung PRIMARY KEY und bei UNIQUE Einschränkungen mit dieser Option nicht angezeigt wurde. NOVALIDATE

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem zusätzliche Kategorien im konvertierten Schema angezeigt wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem das Aktionselement 13185 nicht angezeigt wurde, wenn eine ungelöste Spalte als Argument der CAST Funktion konvertiert wurde.
Teradata	Amazon Redshift	Die Konvertierung von DELETE DELETE ALL AND-Anweisungen zur Verwendung des TRUNCATE Befehls im konvertierten Code wurde verbessert.
Teradata	Amazon Redshift	Die Konvertierung von SET Tabellen wurde verbessert.
Teradata	Amazon Redshift	Verbesserte NORMALIZE Konditionsumwandlung.
Teradata	Amazon Redshift	Der Bewertungsbericht wurde aktualisiert, sodass die Statistiken zur Datenbankschemakonvertierung aus der Liste der Datenbank-Speicherobjekte entfernt wurden.
Teradata	Amazon Redshift	Die Konvertierung der UPDATE Anweisung ohne die FROM Klausel wurde verbessert.
Teradata	Amazon Redshift	Die Unterstützung des VARBYTE Datentyps im konvertierten Code wurde implementiert.
Teradata BTEQ	AWS Glue	Es wurde ein Problem behoben, bei dem die AWS Glue Option „Konvertieren in“ im Kontextmenü deaktiviert war.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem Datentypen im konvertierten Code fehlten.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem Substitutionsvariablen im konvertierten Code falsch in Anführungszeichen gesetzt wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata BTEQ	Amazon- Redshift- RSQL	Ein Problem bei der Konvertierung von Substitutionsvariablen mit Werten in Skripten wurde behoben. FastLoad
Vertica	Amazon Redshift	Die Unterstützung des TIME Datentyps im konvertierten Code wurde implementiert.
Vertica	Amazon Redshift	Verbesserte Konvertierung von ORDER BY Ausdrücken SELECT DISTINCT und.
Vertica	Amazon Redshift	Unterstützung für die Konvertierung von Einschränkungen hinzugefügt.
Vertica	Amazon Redshift	Es wurde ein Fehler behoben, bei dem ein Bewertungsbericht nicht als Datei mit kommagetrennten Werten (CSV) gespeichert wurde.

Versionshinweise für Build 659 AWS SCT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Der Assistent für neue Projekte, der einen kombinierten Bewertungsbericht für mehrere Quelldatenbanken generiert, wurde verbessert.
Alle	Alle	Es wurde ein Problem behoben, bei dem das Erweiterungspaket in Projekten, die mehrere Quell- und Zieldatenbanken enthalten, nicht erstellt wurde.
Alle	Alle	Verbesserte Konvertierung von SQL-Code, der in den Quellcode der Anwendung eingebettet ist.
Alle	Alle	Es wurde die Möglichkeit hinzugefügt, Skripts aus verschiedenen Ordnern in der AWS SCT Befehlszeilenschnittstelle auszuführen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Amazon Redshift	Die Warnmeldung, die angezeigt wird, wenn Benutzer in Migration Projekten mit der virtuellen Zieldatenbankplattform Amazon Redshift die Option Optimierung ausführen wählen, wurde verbessert.
Alle	Aurora PostgreSQL	Unterstützung der PostgreSQL-Hauptversion 13 auf der Aurora PostgreSQL-Compatible Edition als Migrationsziel implementiert.
Alle	Amazon RDS für MySQL	Standardmäßig wurde die Codekonvertierung ohne Berücksichtigung der Groß- und Kleinschreibung implementiert.
Azure Synapse Analytics	Amazon Redshift	Es wurde ein Fehler behoben, bei dem die Verbindung zu einer Quelldatenbank in der Befehlszeilenschnittstelle fehlschlug.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Die Konvertierung von Prozeduren, die UPDATE Anweisungen mit Join-Bedingungen enthalten, wurde verbessert.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Verbesserte Konvertierung von Triggern, gespeicherten Prozeduren und Funktionen, die den Wert nach dem Gleichheitszeichen enthalten .
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Ein Transformationsfehler für Prozeduren mit der DELETE Anweisung und dem OR Operator wurde behoben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Die Konvertierung der OUTPUT Klausel wurde verbessert.
Microsoft SQL Server DW	Amazon Redshift und AWS Glue	Verbesserte Konvertierung des NUMERIC Datentyps.
Microsoft SQL Server DW	Amazon Redshift	Verbesserte Konvertierung von Ansichten, die einen Tabellenalias mit demselben Namen wie die Originaltabelle haben.
Microsoft SSIS	AWS Glue	Es wurde ein Problem behoben, bei dem die AWS Glue Verbindungsinformationen nicht im Fenster Verbindungen konfigurieren angezeigt wurden.
Netezza	Amazon Redshift	Es wurde die Möglichkeit hinzugefügt, die Ausführung von CDC-Datenmigrationsaufgaben (Change Data Capture) täglich zu wiederholen.
Netezza	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Registerkarte „Aufgaben“ inaktiv wurde, nachdem die Registrierung eines Datenextraktionsagenten aufgehoben wurde.
Netezza	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Bestätigung der Registrierung des Datenmigrationsagenten nicht auf der Benutzeroberfläche angezeigt wurde.
Netezza	Amazon Redshift	Es wurde ein Problem behoben, bei dem eine Verbindung zu einer Quelldatenbank mit einem Loader-Fehler fehlschlug.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Netezza	Amazon Redshift	Es wurde ein Fehler behoben, bei dem Datenmigrationsagenten nach dem Öffnen eines gespeicherten Projekts nicht ausgeführt werden konnten.
Oracle	Amazon RDS für Oracle	Unterstützung von Oracle Unified Auditing implementiert.
Oracle	PostgreSQL L Aurora PostgreSQL L	Implementierung der SQL-Code-Konvertierung in C#-Anwendungen. Weitere Informationen finden Sie unter Konvertieren von SQL-Code in C#-Anwendungen .
Oracle	PostgreSQL L Aurora PostgreSQL L	Es wurde eine neue Transformationslogik für Objektnamen implementiert, bei denen die Groß- und Kleinschreibung beachtet wird, um die Sichtbarkeit von Änderungen bei der Codekonvertierung zu verbessern. AWS SCT konvertiert Objektnamen in Großbuchstaben in Kleinbuchstaben. Das Gegenteil ist auch der Fall; AWS SCT konvertiert Objektnamen in Kleinbuchstaben in Großbuchstaben. Andere Objektnamen und reservierte Wörter werden unverändert konvertiert.
Oracle	PostgreSQL L Aurora PostgreSQL L	Verbesserte Konvertierung von Hash-Partitionen ohne NOT NULL Einschränkung.
Oracle	Aurora PostgreSQL L	Unterstützung für die Konvertierung von Oracle CHECKFOREIGN KEY, und NOT NULL Constraints mit der ENABLE NOVALIDATE Klausel hinzugefügt.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem die falschen Werte für Fließkommazahlen migriert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle DW	Amazon Redshift und AWS Glue	Es wurde ein Problem mit leeren Spalten im Bewertungsbericht zur Datenbankmigration in einer Datei mit kommagetrennten Werten (CSV) behoben.
SAP ASE	PostgreSQL Aurora PostgreSQL	Ein Problem mit einer unerwarteten Unterbrechung der Konvertierung wurde behoben.
Snowflake	Amazon Redshift	Die Konvertierung des VARIANT Datentyps wurde verbessert.
Teradata	Amazon Redshift	Verbesserte Konvertierung der COLLECT STATISTICS Anweisung.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem das Aktionselement 9998 nicht angezeigt wurde, wenn verschachtelte Ansichten mit PERIOD Spalten konvertiert wurden.
Teradata	Amazon Redshift und AWS Glue	Es wurde ein Problem behoben, bei dem eine virtuelle AWS Glue Zielplattform nach dem Öffnen eines gespeicherten Projekts nicht in der Benutzeroberfläche angezeigt wurde.
Teradata BTEQ	AWS Glue	Es wurde ein Problem behoben, bei dem die Konvertierung auf eine virtuelle AWS Glue Zielplattform nach dem Öffnen eines gespeicherten Projekts nicht unterstützt wurde.
Teradata BTEQ	Amazon-Redshift-RSQL	Die Syntaxhervorhebung des konvertierten Codes wurde verbessert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata BTEQ	Amazon- Redshift- RSQL	Die Überprüfung der Parameterwerte nach dem Upload wurde implementiert. Nicht unterstützte Werte werden auf der Registerkarte Variablen hervorgehoben.
Vertica	Amazon Redshift	Die Konvertierung von Aggregatfunktionen wurde implementiert.
Vertica	Amazon Redshift	Die Konvertierung von Projektionen in materialisierte Ansichten wurde implementiert und die Benutzeroberfläche verbessert, die den Quellcode von Projektionen anzeigt.

Versionshinweise für AWS SCT Build 658

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Integration mit bereitgestellt AWS Secrets Manager. Sie können jetzt die Anmeldeinformationen für die Datenbankverbindung verwenden, die in Secrets Manager gespeichert sind.
Alle	Alle	Unterstützung für Skripts im YAML-Format in der AWS SCT Befehlszeilenschnittstelle hinzugefügt.
Alle	Amazon Redshift	Unterstützung von Amazon S3 S3-Schnittstellenendpunkten (VPCE) in Datenextraktionsagenten implementiert.
Alle	Amazon Redshift	Unterstützung für die virtuelle Zieldatenbankplattform Amazon Redshift wurde zusätzlich zu dem bereits unterstützten Amazon Redshift und AWS Glue dessen Kombination hinzugefügt.
Greenplum	Amazon Redshift	Es wurde ein Problem behoben, bei dem mit der Option Als SQL speichern der konvertierte SQL-Code nicht in einer Datei gespeichert wurde.
IBM Db2 (LUW)	Aurora MySQL	Verbesserte Konvertierung zur Unterstützung neuer Funktionen der Amazon Aurora MySQL-Compatible Edition mit MySQL 8.0-Kompatibilität.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft Azure SQL Database		
Microsoft SQL Server		
Oracle		
SAP ASE		
Microsoft SQL Server	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Es wurde ein Problem behoben, bei dem der Aktionspunkt 810 bei Bedarf nicht angezeigt wurde.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung von Prozeduren mit UPDATEDELETE, und INSERT -Anweisungen wurde verbessert.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem der Aktionspunkt 7810 bei Bedarf nicht angezeigt wurde.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Die Konvertierung einer EXEC Anweisung, die in einer Anweisung verschachtelt ist, wurde verbessert. IF . . . ELSE
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von indizierten Ansichten.
Netezza	Amazon Redshift	Verbesserte Datenmigrationsagenten durch die Nachverfolgung von Live-Transaktionen bei Volllast im CDC-Vorgang (Change Data Capture). Sie können Datenmigrationsaufgaben jetzt beenden, wenn der Start der CDC-Sitzung zu einem bestimmten Zeitpunkt geplant ist. Außerdem können Sie die Stufe der Fehlerprotokollierung in der Konsole sehen, nachdem Sie eine Aufgabe mit CDC beendet haben.
Oracle	Alle	Der Tabellenlader wurde verbessert, um sicherzustellen, dass Objekte mit Freigabeoptionen AWS SCT geladen werden.
Oracle	Aurora PostgreSQL PostgreSQL	Die Konvertierung der SYSDATE Funktion wurde verbessert und die Möglichkeit hinzugefügt, die Zeitzone in den Konvertierungseinstellungen zu ändern.
Oracle	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem dynamische Anweisungen nicht konvertiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem behoben, bei dem der konvertierte Code keine vom System generierten Namen enthielt.
Oracle Oracle DW	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von SELECT Anweisungen, die innerhalb von Triggern verschachtelt sind, wurde verbessert.
Oracle DW	Amazon Redshift	Die Konvertierung der TO_TIMESTAMP_TZ Funktionen TO_DATE TO_TIMESTAMP , und im Erweiterungspaket wurde verbessert.
Snowflake	Amazon Redshift	Es wurde eine Option hinzugefügt, um konvertierten SQL-Code in verschiedenen Dateien für jedes Objekt oder für jede Anweisung zu speichern.
Teradata	Amazon Redshift	Verbesserte Konvertierung der CONCAT Funktion.
Teradata	Amazon Redshift	Verbesserte Konvertierung einer SELECT Anweisung, die in einer WHERE Klausel verschachtelt ist.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei SET dem MULTISSET Tabellen falsch konvertiert wurden, nachdem Benutzer eine Tabelle gelöscht und neu erstellt hatten.
Teradata	Amazon Redshift	Die Konvertierung der Prozeduren, die eine WITH Klausel enthalten, wurde verbessert.
Teradata	Amazon Redshift	Verbesserte Konvertierung des DATE Datentyps.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem während der FastExport Skriptkonvertierung ein unerwarteter Transformatorfehler auftrat.
Teradata BTEQ	Amazon-Redshift-RSQL	Unterstützung für die Konvertierung eines Join-Index in eine materialisierte Ansicht hinzugefügt.
Teradata BTEQ	Amazon-Redshift-RSQL	Unterstützung für die Konvertierung einer TITLE Definition, die mehrere Zeilen umfasst, wurde hinzugefügt.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem die Größe eines Geodatentyps nicht konvertiert wurde.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem die Parameternamen in Kleinbuchstaben umgewandelt wurden.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem eine gespeicherte Prozedur, die in einer MACRO Anweisung verschachtelt ist, nicht konvertiert wurde.
Vertica	Amazon Redshift	Die Konvertierung des ALL Operators wurde verbessert.
Vertica	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Use Union all view? Option in den Konvertierungseinstellungen nicht angewendet wurde.
Vertica	Amazon Redshift	Verbesserte Konvertierung der TIME WITH TIMEZONE Datentypen TIME und.
Vertica	Amazon Redshift	Ein Problem beim Laden von Flex-Tabellen wurde behoben.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 657

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Apache Log4j wurde auf Version 2.17 aktualisiert, um Sicherheitslücken zu beheben.
Alle	Amazon Redshift	Verbesserte Projekte zur Schemaoptimierung, bei denen wichtige Verwaltungsstatistiken nicht im Projekt gespeichert wurden. AWS SCT
Amazon Redshift	Amazon Redshift	Ein Problem mit der Aktualisierung der Serverinformationen wurde behoben.
Apache Cassandra	Amazon DynamoDB	Ein Problem mit Zuordnungsregeln bei der Verwendung der AWS SCT Befehlszeilenschnittstelle wurde behoben.
Apache Cassandra	Amazon DynamoDB	Es wurde ein Problem behoben, bei dem die Migrationsaufgabe aufgrund eines aktualisierten Titels im Zertifikat nicht erstellt wurde.
Microsoft SQL Server	Aurora PostgreSQL	Es wurde ein Problem behoben, sodass das Aktionselement 7672 bei der Konvertierung von Microsoft SQL Server-Prozeduren mit dynamischem SQL nicht angezeigt wurde.
	PostgreSQL	
Azure SQL Database	Aurora PostgreSQL	Verbesserte Konvertierung von Tabellenwertfunktionen.
Microsoft SQL Server	PostgreSQL	

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem das OUT Argument in einer gespeicherten Prozedur mit dem Standardrückgabewert nicht in das INOUT Argument konvertiert wurde.
Greenplum	Amazon Redshift	Verbesserte Optimierungsstrategien, indem die am häufigsten verwendeten Tabellen und Spalten in der QueryLog Tabelle gefunden wurden.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Probleme bei der Konvertierung der folgenden Dateien wurden behoben: <ul style="list-style-type: none"> • Zuweisungsoperator für Zeichenkettenverkettung () += • SCOPE_IDENTITY Funktion • varchar(max) -Datentyp
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Verbesserte Konvertierung von Ansichten mit nicht unterstützten Funktionen.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Es wurde ein Problem behoben, bei dem nicht unterstützte Funktionen als Argument für eine andere Funktion falsch konvertiert wurden.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Die Konvertierung von Verweisen auf Übergangstabellen wurde verbessert.
Microsoft SQL Server DW	Amazon Redshift	Die Kategorie Aggregatfunktionen wurde dem Metadatenbaum der Quelldatenbank hinzugefügt.
Microsoft SQL Server DW	Amazon Redshift	Verbesserte Konvertierung des TIME Datentyps.
Azure Synapse Analytics Greenplum Netezza Microsoft SQL Server DW Snowflake Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem DROP CREATE Skripts nicht gespeichert wurden, wenn eine virtuelle Zieldatenbankplattform verwendet wurde.
Microsoft SQL Server-Integrationsdienste	AWS Glue	Es wurde ein Problem behoben, bei dem die Skripts von Quellobjekten nicht in der Benutzeroberfläche angezeigt wurden.
Netezza	Amazon Redshift	Verbesserte Optimierungsstrategien durch Auswahl der Faktentabelle und geeigneter Dimensionen für die Kollokation.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Aurora PostgreSQL L PostgreSQL L	Es wurde ein Problem bei der korrekten Konvertierung von Oracle-Triggern behoben, die Sequenznummern verwenden.
Oracle	Aurora PostgreSQL L PostgreSQL L	Die Konvertierung von Ansichten mit öffentlichen Datenbank-Links wurde verbessert.
Oracle DW	Amazon Redshift	Verbesserte Optimierungsstrategien durch Analyse der Kardinalität von Indexspalten.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem benutzerdefinierte benutzerdefinierte Skalarfunktionen mit Zeichenkettenverkettung falsch konvertiert wurden.
Snowflake	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Option Als SQL speichern nicht in der Benutzeroberfläche angezeigt wurde.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Erfassung von Statistiken mit der <code>LOADER ERROR</code> Ausnahme fehlschlug.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Option Bericht erstellen nicht in der Benutzeroberfläche angezeigt wurde.
Teradata	Amazon Redshift	Die Konvertierung der <code>CAST</code> Funktion wurde verbessert.
Teradata	Amazon Redshift	Eine fehlerhafte Konvertierung für <code>ST_Line_Interpolate_Point</code> wurde behoben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Ein unerwarteter Wert wurde aus dem Python-Bibliothekspfad entfernt.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Resolver-Fehler behoben, der bei der Konvertierung mehrerer FastLoad Skripte auftrat.
Teradata BTEQ	Amazon-Redshift-RSQL	Verbesserte Konvertierung der DATABASE Befehls- und Geometrie datentypen.
Teradata BTEQ	AWS Glue	Ein Problem mit einer falschen Synchronisation der Quell- und Zielskripts in der Benutzeroberfläche wurde behoben.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 656

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Unterstützung für mehrere Quell- und Zieldatenbanken innerhalb eines Projekts hinzugefügt. Benutzer können jetzt Zuordnungsregeln erstellen, die unterschiedlichen Datenbankschemas und Zielplattformen im selben Projekt entsprechen.
Alle	Alle	Unterstützung für virtuelle Zieldatenbankplattformen hinzugefügt. Benutzer müssen jetzt keine Verbindung zu einer Zieldatenbank herstellen, um zu sehen, wie ihr Quelldatenbankschema AWS SCT konvertiert wird.
Alle	Alle	Verbesserungen der Benutzeroberfläche: <ul style="list-style-type: none"> •

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		<p>Die Optionen Connect zum Server herstellen und Verbindung zum Server trennen wurden den Quell- und Ziel-Metadatenbäumen hinzugefügt.</p> <ul style="list-style-type: none"> • Es wurde eine Option hinzugefügt, um einen Datenbankserver aus dem AWS SCT Projekt zu entfernen.
Cassandra	Amazon DynamoDB	Es wurde ein Suchproblem behoben, bei dem die CASSANDRA_HOME Variable keinen Schrägstrich (/) hinter <code>cassandra.yaml</code> oder dem <code>conf</code> Ordner enthielt.
Cassandra	Amazon DynamoDB	Unterstützung für Amazon Machine Image (AMI) für Amazon Linux 2 hinzugefügt.
Cassandra	Amazon DynamoDB	Verbesserte Fehlermeldung, die angezeigt wird, wenn ein falscher Schlüssel für Cassandra eingegeben wird.
Cassandra	Amazon DynamoDB	Verbesserte Konvertierung durch Änderung einer Eigenschaft in der <code>cassandra-env.yaml</code> Datei in Abhängigkeit von der Version der Zieldatenbank.
Cassandra	Amazon DynamoDB	Die Java-Version auf dem Ziel-Cassandra Datacenter wurde auf 1.8.0 erhöht.
Greenplum	Amazon Redshift	Verbesserte Optimierungsstrategien in den Projekteinstellungen.
Greenplum	Amazon Redshift	Es wurde ein Datenmigrationsproblem behoben, bei dem Objekte mit diesem Fehler nicht auf die Datenbank angewendet wurden: <code>An I/O error occurred while sending to the backend</code> .

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Greenplum Microsoft SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem die <code>Apply RTRIM to string columns</code> Option nicht in der Benutzeroberfläche angezeigt wurde.
Microsoft SQL Server	Babelfish für Aurora PostgreSQL	Unterstützung für Babelfish for Aurora PostgreSQL als Zielplattform hinzugefügt. Benutzer können jetzt einen Bewertungsbericht erstellen, um die Migration von SQL Server zu Babelfish for Aurora PostgreSQL abzuschätzen.
Netezza	Amazon Redshift	Verbesserte Optimierungsstrategien in den Projekteinstellungen.
SAP ASE	Aurora PostgreSQL PostgreSQL	Es wurde die Möglichkeit implementiert, eindeutige Namen für Indizes zu generieren.
SAP ASE	Aurora PostgreSQL PostgreSQL	Ein Problem mit einer doppelten Indexspalte im Zielskript wurde behoben.
Snowflake	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Optionen Leere Schemas ausblenden, Leere Datenbanken ausblenden und Systemdatenbanken/Schemas ausblenden nicht in der Benutzeroberfläche angezeigt wurden.
Teradata	Amazon-Redshift-RSQL	Unterstützung für die Konvertierung von MultiLoad Teradata-Jobskripten in Amazon Redshift RSQL-Skripts hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Ein Problem mit der Konvertierung von Substitutionsvariablen in und-Skripten wurde behoben. FastLoad FastExport
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem Aktionspunkte nach dem Wechsel von der Registerkarte „Zusammenfassung“ nicht auf der Registerkarte „Aktionspunkte“ angezeigt wurden.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem nach dem Generieren eines Berichts während der FastExport Skriptkonvertierung ein Fehler auftrat.
Teradata	Amazon-Redshift-RSQL	Formatierungsprobleme nach der Konvertierung von Shell-Skripten wurden behoben.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, sodass AI 13177 jetzt im konvertierten Skript kommentiert wird.
Teradata	Amazon Redshift	Eine fehlerhafte Konvertierung von temporalen Tabellen wurde behoben.
Teradata	Amazon Redshift	Die Konvertierung der SET QUERY_BAND Anweisung wurde verbessert.
Teradata	Amazon Redshift	Eine fehlerhafte Konvertierung der NORMALIZE Operation wurde behoben.
Vertica	Amazon Redshift	Die Beschreibung von AI 17008 wurde verbessert.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 655

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, durch das sichergestellt wurde, dass alle Bewertungsprobleme in Berichten angezeigt werden, wenn FastLoad oder verwendet MultiLoad wird.
Teradata	Amazon-Redshift-RSQL	Unterstützung für die Konvertierung von FastExport Teradata-Jobskripten in Amazon Redshift RSQL-Skripts hinzugefügt.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, das sicherstellte, dass die Aktion „Manifest in S3 speichern“ im Offline-Modus aktiviert ist, wenn sie verwendet wird. FastLoad
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, um sicherzustellen, dass Zuordnung sregeln für Skripte wie angewendet werden FastLoad.
Greenplum	Amazon Redshift	Die unterstützte Mindesttreiberversion für Greenplum wurde auf 42.2.5 erhöht.
Greenplum	Amazon Redshift	Es wurde eine Verbindung zu Greenplum über SSL mit der Treiber version 42.2.5 oder höher hinzugefügt.
Oracle DW	Amazon Redshift	Verbesserte Unterstützung für die Ausführung benutzerdefinierter benutzerdefinierter Skalarfunktionen (UDF) innerhalb einer anderen UDF.
Oracle DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem Funktionen mit diesem Fehler nicht auf die Datenbank angewendet wurden: <code>Failed to compile udf</code> .
Oracle DW	Amazon Redshift	Verbesserte Konvertierung durch Verwendung geeigneter Typdeklarationen, z. B. <code>pls-type</code> für <code>%ROWTYPE</code> -Parameter.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem Probleme mit der Bewertung des Informationstyps nicht im Bericht angezeigt wurden.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Transformer-Fehler nach der Konvertierung einiger Skripts behoben.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, sodass ein Problem jetzt im konvertierten Skript kommentiert wird.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem FastExport ->EXPORT -> 'null' statt 'CAST' nach der Konvertierung angezeigt wurde.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem einige Funktionen eines Erweiterungspakets fehlschlugen, wenn die Treiberversion Cause: [JDBC Driver]String index out of range: 0 1.2.43 verwendet wurde
Teradata	Amazon Redshift	SET-Tabellenkonvertierung — SET-Tabellenemulation für Insert-Select-Anweisungen hinzugefügt.
Teradata	Amazon Redshift	CAST — Unterstützt zusätzliches Casting von Datentypen.
Teradata	Amazon Redshift	Eine fehlerhafte Konvertierung für „other_current_time_01“ wurde behoben
Teradata	Amazon Redshift	Teradata FastExport — Amazon Redshift RSQL: Verbesserte Konvertierung von Teradata-Befehlen — Field FastExport
Teradata	Amazon Redshift	Teradata FastExport — Amazon Redshift RSQL: Verbesserte Konvertierung von Teradata-Befehlen — Layout FastExport

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem sich das Zielskript von Objekten mit SAVE EXCEPTIONS STATEMENT nach der erneuten Konvertierung änderte.
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde ein Problem behoben, bei dem nach der <code>proc_cursor_with_calc_columns</code> Konvertierung ein falsches Feld in der ORDER BY-Klausel angegeben wurde.
Oracle	PostgreSQL Aurora PostgreSQL	Behoben: Bei einer ASSOCIATIVE COLLECTION-Konvertierung ist eine zusätzliche <code>aws_oracle_ext\$array_id\$temporary</code> Variablendeklaration erforderlich.
Oracle	PostgreSQL Aurora PostgreSQL	Behoben: Die falsche Konvertierung eines PRIMÄRSCHLÜSSELS mit demselben Namen eines INDEX, der derselben Tabelle gehört.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 654

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL L Aurora PostgreSQL L	Es wurde ein Problem mit dem Fehler beim Parsen von Pseudospalten für hierarchische Abfragen und PRIOR-Spalten behoben.
Oracle	PostgreSQL L Aurora PostgreSQL L	Es wurde ein Problem behoben, bei dem ein mehrzeiliger Kommentar mit einem Schrägstrich und einem Sternchen (/*) korrekt konvertiert wurde.
Oracle	PostgreSQL L Aurora PostgreSQL L	Dem Erweiterungspaket wurde die Systemansicht-Emulation USER_COL_COMMENTS hinzugefügt.
Oracle	PostgreSQL L Aurora PostgreSQL L	Verbesserte Konvertierung von Literalen in Anführungszeichen.
DB2 LUW	PostgreSQL L Aurora PostgreSQL L	Verbesserte Konvertierung von LABEL-Anweisungen, die Bezeichnungen in den Beschreibungen von Tabellen, Ansichten, Aliasnamen oder Spalten hinzufügen oder ersetzen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	None	Die Systemtabelle SYS.USER\$ wurde durch die Ansicht DBA_USERS ersetzt und die Abfragen wurden verbessert.
Oracle DW	Amazon Redshift	Oracle DW-Metadatenabfragen wurden aktualisiert.
Teradata	Amazon-Redshift-RSQL	Unterstützung für die Konvertierung von Shell-, Teradata- und Teradata FastLoad Basic Teradata Query (BTEQ) -Skripts in Amazon Redshift RSQL-Skripts hinzugefügt.
Teradata BTEQ	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem „merge_01“ falsch konvertiert wurde.
Teradata BTEQ	Amazon-Redshift-RSQL	Das Problem, dass End of Identify (EOI) am Ende eines Skripts in einer neuen Zeile angezeigt wird, wurde behoben.
Azure Synapse	Amazon Redshift	Die Fehlermeldung, die angezeigt wird, wenn ein falsches Passwort für Azure Synapse eingegeben wurde, wurde verbessert.
Teradata	Amazon Redshift	Die Konvertierung von UPDATE-Anweisungen wurde verbessert, sodass der richtige Aliasname gemäß dem Teradata-Standard übernommen wird.
Teradata	Amazon Redshift	Es wurde ein Fehler bei der Cursorkonvertierung behoben, bei dem keine Aktionen empfangen wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem bei einer TD_NORMALIZE_OVERLAP-Konvertierung Zeilen gelöscht wurden.
Teradata	Amazon Redshift	Unterstützt jetzt die strikte Datumsprüfung für die erweiterte TO_DATE-Funktion.
Teradata	Amazon Redshift	Verbesserte Konvertierung der integrierten Funktion TO_NUMBER (n).

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Kategorie Schemas in der Metadatenstruktur fehlte.
Greenplum	Amazon Redshift	Die Auswahl GP_SEGMENT_ID wurde zur Liste hinzugefügt, wenn eine virtuelle Partition für eine Greenplum-Tabelle erstellt wurde.
Greenplum	Amazon Redshift	Es wurde ein Problem behoben, bei dem Funktionen nicht auf das Ziel angewendet wurden.
MS SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem nach der Konvertierung ohne AI 9996 ein Transformationsfehler auftrat.
MS SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem beim Öffnen des Erweiterungs paket-Assistenten ein Fehler protokolliert wurde.
MS SQL Server DW	Amazon Redshift	Es wurde ein Problem behoben, bei dem ein falscher Kommentarstil für Redshift-Python-Funktionen verwendet wurde.
Netezza	Amazon Redshift	Es wurde ein Problem behoben, bei dem ein Netezza-Redshift-Erweiterungspaket mit einem Profil nicht erstellt werden konnte. AWS
Teradata	Amazon-Redshift-RSQL	Die Konvertierung des Befehls SESSIONS wurde verbessert. FastLoad
Teradata	Amazon-Redshift-RSQL	Verbesserte Bewertungsberichte für FastLoad Skripts.
Teradata	Amazon-Redshift-RSQL	FastLoad Die WRITER-Aktion „In S3 speichern“ wurde implementiert.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem die Schaltflächen Skript FastLoad speichern\ Manifest in S3 speichern nicht aktiv waren.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem FastLoad multifile_script nach der Konvertierung nur eine Manifestdatei anstelle der erwarteten drei Dateien erstellte.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem FastLoad zusätzliche Ordner in einem S3-Pfad angezeigt wurden.
Teradata	Amazon-Redshift-RSQL	Es wurde ein Problem behoben, bei dem der falsche Name der Manifestdatei in einem S3-Pfad angegeben FastLoad war.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 653

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL Aurora PostgreSQL	Es wurde die Fähigkeit implementiert, dynamisches SQL zu konvertieren, das in aufgerufenen Funktionen oder Prozeduren erstellt wurde.
Oracle	PostgreSQL Aurora PostgreSQL	Verbesserte dynamische SQL-Konvertierung: In-Parameter als Bind-Variablen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle DW 18, 19	Amazon Redshift	Verbesserungen bei der Konvertierung von Oracle zu Redshift implementiert: verbesserte integrierte Konvertierungsfunktionen. Aggregiertes LISTAGG; Analytisches LISTAGG.
Oracle DW 18,19	Amazon Redshift	Verbesserungen bei der Konvertierung von Oracle zu Redshift implementiert: Neue Funktionen abfragen.
Vertica	Amazon Redshift	Verbesserungen bei der Konvertierung von Vertica nach Redshift implementiert: SSL-zu-JDBC-Verbindung mit SSL=True.
MS SQL Server DW	Amazon Redshift	Verbesserungen bei der Konvertierung von MS SQL Server nach Redshift: Externe Tabellen.
Teradata	Amazon Redshift	Verbesserungen bei der Konvertierung von Teradata nach Redshift: Interval-Datentypen, arithmetische Operationen.
Teradata	Amazon Redshift	Verbesserungen bei der Konvertierung von Teradata nach Redshift: Support für laterale Spaltenalias.
Oracle	None	Die folgenden Loader-Abfragen verwenden jetzt anstelle von: DBA_USERS SYS.USER\$ <ul style="list-style-type: none"> • get-tree-path-list- by-name-path .sql • estimate-table-or-view- .sql constraints-by-schema • estimate-table-or-view- .sql constraints-by-selected-schemas
Teradata	Amazon Redshift	Verbesserte Ausrichtung von Kommentaren, wenn SCT Teradata-Makros in gespeicherte Redshift-Prozeduren konvertiert.
Oracle DW	Amazon Redshift	Verbesserte Konvertierung von Date/Timestamp-Formatelementen: TO_DATE, und TO_TIMESTAMP TO_TIMESTAMP_TZ
Teradata	Amazon Redshift	Der Fehler bei der Teradata-Cursorkonvertierung wurde behoben.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Es wurde ein Problem behoben, das dazu führte, dass Attribute von TD_NORMALIZE_OVERLAP während der Konvertierung gelöscht wurden.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die MAX Funktion ignoriert wurde, als SCT eine Abfrage konvertierte.
Teradata	Amazon Redshift	SCT konvertiert jetzt die Teradata CHARACTERS-Funktion in die Redshift LENGTH-Funktion.
Teradata	Amazon Redshift	SCT unterstützt jetzt die Konvertierung von FORMAT in TO_CHAR für die am häufigsten verwendeten Formate.
Alle	Alle	Verbesserte Konvertierung verschlüsselter Routinen.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 652

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	PostgreSQL	App-Sperren für sp_getapplock und sp_releaseapplock Funktionen hinzugefügt.
None	Amazon Redshift	Verbesserung der Befehlszeilenschnittstelle (CLI): Implementierter Skriptbefehlsmodus.
Oracle	PostgreSQL	Das Sampling von Routineparametern wurde in dynamischem SQL implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
	Aurora PostgreSQL L	
Oracle	PostgreSQL L Aurora PostgreSQL L	Verbesserungen bei der Konvertierung von dynamischem SQL, das in aufgerufenen Funktionen oder Prozeduren erstellt wurde.
Microsoft SQL Server Oracle DB2 LUW	Aurora PostgreSQL L	Jede Lambda-Funktion wird nur einmal über eine Richtlinie bereitgestellt und konfiguriert, und gängige Lambda-Funktionen werden für alle möglichen Quellen wiederverwendet.
DB2 LUW	PostgreSQL L	Es wurde ein Problem behoben, das zu der Fehlermeldung „9996 — Schweregrad kritisch — Transformer-Fehler ist aufgetreten“ führte, wenn DB2 LUW als Quelle verwendet wurde.
Teradata	Amazon Redshift	Support für rekursive Tabellenausdrücke bei der bevorstehenden Markteinführung von Amazon Redshift.
Azure Synapse	Amazon Redshift	Regeln zur Schemaoptimierung implementiert.
Teradata	Amazon Redshift	Support Zeitzonekonvertierung von Teradata-Makros in gespeicherte Redshift-Prozeduren.
Teradata	Amazon Redshift	Support Arithmetik für PERIOD-Werte.
Teradata	Amazon Redshift	Support die Konvertierung von Teradata Recursive Common Table Expressions (RECURSIVE CTE).

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Support Kennungen mit Berücksichtigung von Groß- und Kleinschreibung über die Benutzereinstellung, <code>enable_case_sensitive_identifier</code> . „COLUMN_NAME“ und „Column_Name“ werden also zu unterschiedlichen Spaltennamen.
Teradata	Amazon Redshift	Das Problem mit dem Datentyp Dezimal wurde behoben, sodass Dezimalfelder mit derselben Genauigkeit konvertiert wurden.
Teradata	Amazon Redshift	Es wurde ein Problem mit der arithmetischen Intervallkonvertierung behoben, sodass die Intervallarithmetiksubtraktion korrekt konvertiert wurde.
Teradata	Amazon Redshift	Die Umwandlung vom Typ Teradata NUMBER in DATE wurde verbessert.
Teradata	Amazon Redshift	Die Umwandlung von Teradata vom Typ DATE in NUMBER wurde verbessert
Teradata BTEQ	Amazon Redshift	Verbesserte Konvertierung des PERID-Datentyps.
Teradata	Amazon Redshift	Es wurde ein Problem beim Laden von Metadaten für eine Tabelle mit GEOMETRY-Spalten behoben, sodass sie jetzt korrekt aus Teradata geladen wurden.
Teradata	Amazon Redshift	Support die Konvertierung von Merge-Anweisungen bei der Konvertierung von Teradata-Makros in gespeicherte Redshift-Prozeduren.
Teradata	Amazon Redshift	Verbesserte Konvertierung einfacher Makros bei der Migration von Teradata zu Redshift.
Teradata	Amazon Redshift	Es wurde sichergestellt, dass bei der Konvertierung der Teradata UPDATE-Anweisungen der richtige Aliasname gemäß dem Teradata-Standard übernommen wurde.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 651

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Verbesserte AWS SCT Berichte zur Aktualisierung der Links zu den aufgeführten empfohlenen Konversionsaktionselementen.
MS SQL Server	PostgreSQL	Unterstützung für die Konvertierung von STR() Funktionen hinzugefügt.
MS SQL Server	PostgreSQL	Unterstützung für die Konvertierung des bitweisen EXOR-Operators (^in Microsoft SQL Server) in PostgreSQL als Operator hinzugefügt. #
Oracle	PostgreSQL	Es wurde ein Problem behoben, bei dem die AWS SCT <code>aws_oracle_ext.UNISTR(null)</code> Erweiterungspack-Funktion NULL auf einem PostgreSQL-Ziel hängen blieb. AWS SCT behandelt jetzt das NULL
Teradata BTEQ	Amazon-Redshift-RSQL	Konvertierungsverbesserungen zur Behebung eines Problems, bei dem die Konvertierung von Amazon Redshift RSQL MERGE zu einem Transformationsfehler führte.
Oracle DW	Amazon Redshift	Verbesserte integrierte Funktionen implementiert.
Oracle DW	Amazon Redshift	Es wurden Verbesserungen hinzugefügt, die auf Funktionen für Metadaten basieren, darunter automatische Listenpartitionierung (TBL_PART_LIST_AUTO), Liste mit mehreren Spalten (TBL_PART_MULTI_LIST) und Interval-Referenz (TBL_PART_RANGE_INTERVAL_REF).

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Keine	Amazon Redshift	Die Beschränkungen der UNION ALL Partitionstabellen für physische Partitionen, die für Konvertierungen verwendet werden, wurden erhöht.
Teradata	Amazon Redshift	Im Rahmen der Bewertungsberichte wurden Verbesserungen bei der Konvertierung vorgenommen.
Teradata	Amazon Redshift	Konvertierungsverbesserungen bei komplexen Teradata MACRO-Konvertierungen.
Teradata	Amazon Redshift	Verbesserte Konvertierung von Teradata-Makros in gespeicherte Amazon Redshift Redshift-Prozeduren beim Auskommentieren von nicht unterstütztem SQL.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die Konvertierung von Teradata-Makros in gespeicherte Amazon Redshift Redshift-Prozeduren zu falschen Aliasnamenreferenzen führte.
Teradata	Amazon Redshift	Die Konvertierung der Teradata-Anweisung wurde verbessert. QUALIFY
Teradata	Amazon Redshift	Die Konvertierung wurde verbessert, um Kommentare an Amazon Redshift zu übertragen und einen Verlauf der an der Ansicht vorgenommenen Änderungen beizubehalten.
Teradata	Amazon Redshift	Es wurde ein Problem behoben, bei dem die RESET WHEN-Klausel nicht zur korrekten Konvertierung führte.
Teradata BTEQ	Amazon Redshift	Verbesserte Konvertierung von BTEQ-Skripten, die MERGE-Anweisungen enthalten.
Teradata	Amazon Redshift	Es wurden integrierte Funktionen hinzugefügt, um die Konvertierung von Feldern vom Datentyp PERIOD zu verbessern.
Microsoft SQL Server	Amazon Redshift	Verbesserte Zuordnung von Transformationsdatentypen für den TIME-Datentyp.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Zugriff auf die Erstveröffentlichung des AWS Schema Conversion Tool CLI-Referenzhandbuchs im PDF-Format hinzugefügt. Siehe AWS Schema Conversion Tool CLI-Referenz .

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 650

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	<p>Aktualisierte und erweiterte Verwendung von Extraktor-Agenten, einschließlich:</p> <ul style="list-style-type: none"> • Eine Konfiguration für die Verwendung mit gemeinsam genutztem Speicher und einem dedizierten Kopieragenten. • Exportieren und Importieren von Datenextraktionsaufgaben von einem Projekt in ein anderes. • Support für Azure SQL Data Warehouse (Azure Synapse) als Quelle. • Verwendung der nativen Netezza-Partitionierung. <p>Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift.</p>
Alle	Amazon RDS PostgreSQL L 13	AWS SCT unterstützt jetzt Amazon RDS PostgreSQL 13 als Ziel.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Aurora PostgreSQL	Die Konvertierung einer Ergebnismenge von einer Microsoft SQL Server-Prozedur in ein Aurora PostgreSQL-Ziel wurde verbessert.
Oracle DW	Amazon Redshift	Verbesserungen bei der Konvertierung von Oracle zu Amazon Redshift implementiert.
Oracle DW	Amazon Redshift	Verbesserungen bei der Konvertierung dynamischer SQL-Anweisungen wurden implementiert.
Oracle DW	Amazon Redshift	Verbesserungen an der SQL UDF-Konvertierung implementiert.
Oracle DW	Amazon Redshift	Die Meldung, die die Konvertierung von EXTERNEN TABELLEN AWS SCT nicht unterstützt, wurde geklärt.
Oracle DW	Amazon Redshift	Verbesserte integrierte Konvertierungsfunktionen.
Teradata BTEQ	Amazon-Redshift-RSQL	Verbesserte Handhabung von Substitutionsparametern in BTEQ-Skripten bei Verwendung der GUI. AWS SCT
Microsoft SQL Server DW	Alle	Die mindestens unterstützte JDBC-Treiberversion für Microsoft SQL Server, Azure, Azure Synapse wurde aktualisiert.
Microsoft SQL Server		
Azure		
Azure Synapse		

Behobene Probleme:

- Teradata: Zusätzliche Verbesserungen bei der Makrokonvertierung [BEHOBEN]
- Sonderzeichen fehlten im Ziel, was zu SQL-Fehlern führte. Sie mussten überarbeitet werden, um sie wieder einzufügen [BEHOBEN]
- Allgemeine Verbesserungen

Versionshinweise für AWS SCT Build 649

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server DW	Amazon Redshift	Verbesserungen bei der Konvertierung von MSSQL zu Amazon Redshift zur Unterstützung temporaler Tabellen.
Oracle DW	Amazon Redshift	<p>Integrierte Funktionserweiterungen wurden implementiert, wie z. B.:</p> <p>Konvertierungs-Funktionen</p> <ul style="list-style-type: none"> • TO_BINARY_DOUBLE • TO_BINARY_FLOAT • TO_NUMBER • TO_DATE • TO_TIMESTAMP • TO_TIMESTAMP_TZ • TO_DSINTERVAL • TO_YMINTERVAL • VALIDATE_CONVERSION

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle DW	Amazon Redshift	<p>Funktionserweiterungen für Approximate Query Processing wurden implementiert, wie z. B.:</p> <p>Aggregationsfunktionen</p> <ul style="list-style-type: none"> • ANY_VALUE • APPROX_COUNT_DISTINCT • APPROX_COUNT_DISTINCT_DETAIL • APPROX_COUNT_DISTINCT_AGG • LISTAGG • ZU_APPROX_COUNT_DISTINCT
Teradata	Amazon Redshift	<p>Konvertierungsverbesserungen für die auto Sortierung und die Auswahl der Verteilungsschlüssel von Teradata wurden implementiert. Die DB-Engine wählt automatisch Verteilungs- und Sortierschlüssel aus. Es wurde ein Optionsfeld mit der Bezeichnung Automatische Tabellenoptimierung von Amazon Redshift verwenden, um Einstellungen für aktuelle Projekte > Optimierungsstrategien > Anfängliche Strategie für die Schlüsselauswahl zu aktivieren, eingeführt.</p>
Teradata	Amazon Redshift	<p>Der AWS SCT Tabellenlader wurde verbessert, um sicherzustellen, dass alle Tabellen aus Teradata AWS SCT geladen werden.</p>
Teradata	Amazon Redshift	<p>Konvertierungsverbesserungen wurden implementiert, sodass Amazon Redshift korrelierte Unterabfragemuster unterstützt, die eine einfache WHERE NOT EXISTS-Klausel enthalten.</p>
Teradata	Amazon Redshift	<p>Unterstützung für die Verwendung von ECHO-Befehlen in Makros hinzugefügt.</p>

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
DB2 LUW	PostgreSQL	Unterstützung für die Konvertierung von DYNAMIC RESULTS SETS implementiert, einschließlich:
	Aurora PostgreSQL	<ul style="list-style-type: none"> • Cursorklausel WITH RETURN/WITH RETURN TO CLIENT • Konvertierung von DYNAMIC RESULT SETS-Routinen
Microsoft SQL Server	Aurora PostgreSQL	Unterstützung für aktuelles Aurora RDS PostgreSQL als Ziel implementiert.
Oracle		
DB2 LUW		
SAP ASE		
Microsoft SQL Server	MariaDB	Unterstützung für MariaDB 10.5 als Ziel implementiert.
Oracle		
DB2 LUW		
SAP ASE		
Microsoft SQL Server	MariaDB	Implementierte Unterstützung von INSERT-RETURNING, die eine Ergebnismenge der eingefügten Zeilen zurückgibt.
Oracle	Aurora PostgreSQL	Unterstützung der XMLFOREST-Funktion für die Konvertierung von Oracle nach Aurora PostgreSQL hinzugefügt.

Behobene Probleme:

- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 648

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL L Amazon Aurora PostgreSQL L-Compatible Edition	Benutzerdefinierter Anwendungsmodus für das Aurora PostgreSQL L-Erweiterungspaket implementiert: Operatoren für numerische/ Datums- und Texttypen.
Oracle Microsoft SQL Server DB2 LUW	Aurora PostgreSQL L	Aurora PostgreSQL Lambda Invoke-Konfiguration implementiert: Erstellung der aws_lambda-Erweiterung; IAM-Rollenzuweisung an den Aurora PostgreSQL-Cluster. <ul style="list-style-type: none"> • Oracle — WebAgent E-Mails, Jobs, Warteschlangen, Dateien • DB2 — E-Mails, Aufgaben, Dateien • Microsoft SQL Server — E-Mails, Agent
Oracle	PostgreSQL L	Das Refactoring zur Konvertierung von FORALL-Anweisungen wurde implementiert: <ul style="list-style-type: none"> • FORALL-Anweisung • FÜR ALLE... AUSNAHMEN SPEICHERN • RÜCKKEHR MIT BULK COLLECT

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		<ul style="list-style-type: none"> • SQL%BULK_EXCEPTIONS-Systemsammlung
Oracle DW 18, 19	Amazon Redshift	Verbesserungen bei der Konvertierung von Oracle zu Amazon Redshift implementiert: verbesserte integrierte Konvertierungsfunktionen. Aggregiertes LISTAGG; Analytisches LISTAGG.
Oracle DW 18,19	Amazon Redshift	Verbesserungen bei der Konvertierung von Oracle zu Amazon Redshift implementiert: Neue Funktionen abfragen.
Vertica	Amazon Redshift	Verbesserungen bei der Konvertierung von Vertica zu Amazon Redshift implementiert: SSL-zu-JDBC-Verbindung mit SSL=True.
Microsoft SQL Server DW	Amazon Redshift	Verbesserungen bei der Konvertierung von Microsoft SQL Server nach Redshift: Externe Tabellen.
Teradata	Amazon Redshift	Verbesserungen bei der Konvertierung von Teradata nach Redshift: Interval-Datentypen, arithmetische Operationen.
Teradata	Amazon Redshift	Verbesserungen bei der Konvertierung von Teradata nach Redshift: Support für laterale Spaltenaliase.

Behobene Probleme:

- Allgemeine Verbesserungen

Versionshinweise für AWS SCT Build 647

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	Microsoft SQL Server	RDS unterstützt jetzt die Datenbank-Mail-Funktion.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SQL Server	MySQL	<p>Implementierung des maximalen Namens für jeden Bezeichnertyp — Die maximale Länge von Objektnamen (z. B. Tabellen, Einschränkungen, Spalten) in SQL Server beträgt 128 Zeichen. Die maximale Länge von Objektnamen in MySQL beträgt 64 Zeichen. Um konvertierte Objekte in die MySQL-Datenbank zu schreiben, müssen Sie ihre Namen kürzen. Um doppelte Namen nach dem Ausschneiden zu vermeiden, müssen Sie den neuen Namen eine „Prüfsumme“ des ursprünglichen Objektnamens hinzufügen.</p> <p>Schneiden Sie Namen, die länger als 64 Zeichen sind, wie folgt aus:</p> <pre>[first N chars]() + "" + [checksum]()</pre> <pre>[first N chars] = 64 - 1 - [length of checksum string]</pre> <p>Beispielsweise:</p> <pre>example_of_a_test_schema_with_a_name_length_greater_than_64_characters ?? example_of_a_test_schema_with_a_name_length_greater_than_64_9703</pre>
Oracle	MySQL/Aurora MySQL	Das Laden und Konvertieren von Kommentaren zu Speicherobjekten wurde implementiert. Zum Beispiel die Verarbeitung von Kommentaren zu Tabellen und die Verarbeitung von Kommentaren zu Tabellen-/View-Spalten.
Teradata	Amazon Redshift	Unterstützung für die TIME-Datentypkonvertierung hinzugefügt.
Teradata	Amazon Redshift	Verbesserungen bei der Konvertierung — TD_NORMALIZE_OVERLAP implementiert.
Microsoft SQL Server DW	Amazon Redshift	Verbesserungen bei der Konvertierung — SELECT mit WITH-Klausel; SELECT ohne FROM

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	AWS SCT Data Migration Service Assessor (DMSA) — Mit dieser neuen Funktion können Sie mehrere Server bewerten und einen zusammenfassenden Bericht erhalten, der die beste Zielrichtung für Ihre Umgebung aufzeigt.
Alle	Alle	AWS SCT Assistent — Der Zielvergleich zeigt jetzt Unterschiede zwischen Zielen in einer einzigen Tabellenansicht.
Alle	Alle	Tree Filter UI — Der neu gestaltete Metadatenfilter verarbeitet komplexere Filtermuster.
Alle	Alle	Bewertungsbericht — Der neu gestaltete Abschnitt mit der Warnung bietet eine bessere Beschreibung und ein klareres Verständnis eines Problems.

Behobene Probleme:

- Allgemeine Verbesserungen
- Datenextraktoren — Die Unteraufgabe ist mit `ConcurrentModificationException` [RESOLVED] fehlgeschlagen.
- Microsoft SQL Server zu MySQL — maximale Identifikationslänge [AUFGELÖST].

Versionshinweise für AWS SCT Build 646

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL	Verbesserte Implementierung des Modells im TM-Format.
Oracle	PostgreSQL	Die Maskenimplementierung im SP-Format bietet grundlegende Unterstützung für das SP-Suffix, nur für die englische Sprache.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL	Behandlung langer Oracle-Objektnamen — verarbeitet AWS SCT nun lange Oracle-Objektnamen entsprechend dem Attribut für die maximale Länge des Zielbezeichners.
	Amazon Redshift	Amazon Redshift Redshift-Kodierung AZ64 mit AWS SCT — Kompressionskodierung AZ64 für einige Datentypen hinzugefügt
Teradata	Amazon Redshift	Unterstützung für die Konvertierung impliziter Transaktionen hinzugefügt.
Teradata	Amazon Redshift	Unterstützung für die integrierten Geodatenfunktionen von Teradata hinzugefügt: Methoden <code>ST_LineString</code>
Greenplum	Amazon Redshift	Greenplum-Sequenzkonvertierung — Die nächsten Elemente wurden zu den Eigenschaften-Tabs hinzugefügt: Minimalwert, Maximalwert, Inkrement, Zyklus.
Greenplum	Amazon Redshift	Resolver — Die Auflösung des Datentyps „char“ wurde hinzugefügt.
Greenplum	Amazon Redshift	Länge der Zeichenkonvertierung — Die PL/pgSQL-Konvertierung für den Zeichentyp wurde aktualisiert.
Greenplum	Amazon Redshift	Es wurde ein Problem mit der Auswahl des Greenplum-Verteilungsschlüssels behoben, bei dem eine Tabelle zwar über einen VERTEILUNGSSCHLÜSSEL verfügte, die Tabelle aber nicht als AWS SCT ZUFÄLLIG VERTEILT erkannt und abgerufen werden konnte.
Teradata	Amazon Redshift	Teradata-Cursorunterstützung — Unterstützung für die Cursorkonvertierung hinzugefügt.
Teradata	Amazon Redshift	Identitätsspalten — Unterstützung für die Konvertierung von Identitätsspalten hinzugefügt.
Teradata	Amazon Redshift	INTERVAL-Datentypen — Unterstützung für die Konvertierung von INTERVAL-Datentypen wurde hinzugefügt.

Behobene Probleme:

- Allgemeine Verbesserungen
- Greenplum: Die Konvertierung konnte aufgrund des Fehlers im Protokoll [BEHOBEN] nicht ausgeführt werden.
- MSSQL — PostgreSQL: Transformer-Fehler bei der Konvertierung der LAG-Funktion [BEHOBEN].
- MSSQL — PostgreSQL: SCOPE_IDENTITY [GELÖST].
- AWS SCT hängt in DW-Projekten [BEHOBEN].
- Benötigen Sie eine Zuordnungsregel, um zusätzliches Leerzeichen im Spaltennamen in AWS SCT [RESOLVED] zu entfernen.

Versionshinweise für AWS SCT Build 645

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	Stellen Sie eine Lösung zur Lösung von Teradata-Ansichten bereit, die nicht vollständig qualifiziert sind (Namen der Ansichten oder nicht vollständig qualifizierte Objekte in der Ansicht).
Teradata	Amazon Redshift	Unterstützung der ASCII-Funktion zur Berechnung von Knoten hinzugefügt.
Teradata	Amazon Redshift	Wenn AWS SCT Multibyte-Daten in einer Teradata erkannt werden, die als CHAR definiert sind <code>CHAR(N)</code> , werden sie in Amazon Redshift <code>VARCHAR(3*N)</code> in diese umgewandelt.
Teradata	Amazon Redshift	Stellen Sie eine Teradata-Konvertierung zwischen Daten CAST und Zahlen bereit. <ul style="list-style-type: none"> • <code>SELECT Cast('2020-07-17' AS BIGINT)</code> • <code>SELECT Cast(20200630 - 19000000 AS DATE)</code>
Teradata	Amazon Redshift	Support die Konvertierung von PERIOD Teradata-Datentypen in zwei Amazon Redshift TIMESTAMP Redshift-Spalten:

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		<ul style="list-style-type: none"> PERIOD(TIMESTAMP) PERIOD(TIMESTAMP WITH TIMEZONE)
Teradata	Amazon Redshift	Support Konvertierung der RANK Teradata-Funktion mit RESET WHEN Klausel.
Teradata	Amazon Redshift	Verbesserte Unterstützung von CAST bei expliziten Datentypkonvertierungen und impliziten CASTs für Ausdrücke.
Teradata	Amazon Redshift	Nicht unterstützte korrelierte Unterabfragemuster melden. Weitere Informationen finden Sie unter Korrelierte Unterabfragen im Amazon Redshift Database Developer Guide.
Keine	Amazon Redshift	Verbesserte Tabellen schränken die Unterstützung für RA3-Knoten ein.
Teradata	Amazon Redshift	Unterstützung für die Extraktion von Teradata-Geodaten hinzugefügt. Weitere Informationen finden Sie unter Abfragen räumlicher Daten in Amazon Redshift im Amazon Redshift Database Developer Guide .
Microsoft SQL Server	PostgreSQL	Die Option wurde hinzugefügt, <code>convert_procedures_to_function</code>

Behobene Probleme:

- Allgemeine Verbesserungen

Versionshinweise für AWS SCT Build 644

Die Änderungen für die AWS SCT Versionen 1.0.643 wurden in der Version AWS SCT 1.0.644 zusammengeführt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	<p>Mehrere Konvertierungsverbesserungen.</p> <ul style="list-style-type: none">• Verbesserte Konvertierungen QUALIFY mit dem Tabellenalias.• Verbesserte Konvertierungen mit dem IN Operator.• Verbesserte Konvertierung mit dem LIKE Operator.• Verbesserte Konvertierungen mit Hervorhebung von Problemen im konvertierten Code.• Verbesserte Konvertierungen mit ungewöhnlicher Reihenfolge von WHERE QUALIFY Klauseln in SQL.• Es wurden Transformatorfehler behoben, die während der JOIN() Konvertierung der Prozedur UPD_FT_SVC_TRANS_BH_CBH_IND aufgetreten sind.• Die Konvertierung von Makros in gespeicherte Prozeduren wurde verbessert. <p>Es wurden spezielle AWS SCT CLI-Befehle hinzugefügt, die die bereitgestellten sql/bteq-Skripte analysieren und einen Bericht über die Anzahl der im Quellcode gefundenen Syntaxstrukturen generieren können.</p> <ul style="list-style-type: none">• Anzahl der BTEQ-Befehle• Anzahl der HANDLER• Anzahl der CAST-Fälle• Anzahl der DML/DDI-Fälle•

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		<p>Anzahl der DMLs in aktualisierbaren Ansichten</p> <p>Ein Aktionselement für den Bewertungsbericht wurde hinzugefügt: Teradata-Spalten mit benutzerdefinierten Datumsformaten werden in Amazon Redshift nicht unterstützt.</p>
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Es wurde eine Funktion zum Speichern von Installationsskripten für Erweiterungspakete hinzugefügt.</p> <p>Schweregrad für AI 5334 geändert.</p> <p>Die Leistung bei der Verwendung eines Datensatzes als Paketvariable <code>IMPLEMENTATION</code> wurde verbessert.</p> <p>Unterstützung für <code>XMLAGG</code> Aggregationsfunktionen hinzugefügt</p>
IBM Db2	PostgreSQL/Aurora PostgreSQL	Das Laden und Konvertieren von Kommentaren zur Implementierung von Speicherobjekten wurde hinzugefügt.
MS SQL DW	Amazon Redshift	<p>Verbesserung der Konvertierung: Problem mit <code>beholdenPATINDEX</code>.</p> <p>Verbesserungen der Benutzeroberfläche:</p> <ul style="list-style-type: none"> Für die Implementierung des Quellbaums als SQL speichern. Zusätzliche Logik zur Skriptgenerierung für mehrere Dateien hinzugefügt.
Vertica	Amazon Redshift	Verbesserung der Benutzeroberfläche: Als SQL für die Implementierung des Quellbaums speichern.

Behobene Probleme:

- Allgemeine Verbesserungen bei Konvertierungen zwischen Teradata und Amazon Redshift

- Allgemeine Fehlerkorrekturen und Verbesserungen der Benutzeroberfläche

Versionshinweise für AWS SCT Build 642

Änderungen für AWS Schema Conversion Tool Version 1.0.642.

Note

AWS Schema Conversion Tool (AWS SCT) Die Änderungen in Build 1.0.642 gelten für Windows, Ubuntu und Fedora. Es gibt keinen 1.0.642-Build für macOS.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft SSIS	AWS Glue	Die Konvertierung von Microsoft SQL Server Integration Services (SSIS) ETL-Paketen in wurde implementiert. AWS Glue Weitere Informationen finden Sie unter SSIS konvertieren inAWS Glue mit AWS SCT .
Oracle	MariaDB/ SQL- Modus= Oracle/ MySQL/ Amazon Aurora MySQL	Der PL/SQL-Deklarationsabschnitt wurde in der WITH-Klausel implementiert.
Oracle	PostgreSQL/Aurora PostgreSQL	Unterstützung für <code>DBMS_SESSION.RESET_PACKAGE</code> und <code>DBMS_SESSION.MODIFY_PACKAGE</code> hinzugefügt.
Vertica	Amazon Redshift	Ermöglichen Sie den Export von SQL-Skripts aus einer Vertica-Datenbank nach Amazon Redshift.

Behobene Probleme:

- Verbesserung des Bewertungsberichts.
- Verbesserung der Benutzeroberfläche des Bewertungsberichts.
- Fügen Sie die Möglichkeit hinzu, die JVM-Einstellungen über die Benutzeroberfläche zu ändern.
- Allgemeine Verbesserungen.

Versionshinweise für AWS SCT Build 641

Änderungen für AWS Schema Conversion Tool Version 1.0.641.

Note

AWS Schema Conversion Tool (AWS SCT) Die Änderungen in Build 1.0.641 gelten für Windows, Ubuntu und Fedora. Es gibt keinen 1.0.641-Build für macOS.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle/ MS SQL/ MySQL/ PostgreSQL/ DB2 LUW	Alle	Erstellen Sie Berechnungen des Zeitberichts in der CSV-Datei.
Teradata	Amazon Redshift	Unterstützung für die CSUM-Funktion hinzugefügt. Unterstützung für Geodatentypen von Teradata hinzugefügt.
Teradata	Alle	Unterstützung für die Konvertierung von IDENTITY-Spalten hinzugefügt.
Greenplum	Amazon Redshift	Unterstützung für den Verteilungsstil AUTO während der Greenplum-Tabellenkonvertierung hinzugefügt.
SAP ASE	Alle	Erstellen Sie Berechnungen des Zeitberichts in der CSV-Datei.

Gelöst:

- Verschiedene Fehlerbehebungen.
- Verschiedene Leistungsverbesserungen.

Versionshinweise für AWS SCT Build 640

Die Änderungen für die AWS SCT Versionen 1.0.633, 1.0.634, 1.0.635, 1.0.636, 1.0.637, 1.0.638, 1.0.639 und 1.0.640 wurden in der Version 1.0.640 zusammengeführt. AWS SCT

Note

AWS SCT Die Änderungen in Build 1.0.640 gelten für Windows, Ubuntu und Fedora. Sie gelten nicht für macOS.

Sie können AWS SCT Version 1.0.640 oder höher nicht auf Apple macOS installieren.

AWS SCT Version 1.0.632 war die letzte Version, die die Installation auf Apple macOS unterstützte.

In den folgenden Tabellen finden Sie eine Liste der Funktionen und Fehlerbehebungen für die AWS Schema Conversion Tool -Versionen, die in Version 1.0.640 kombiniert wurden. Diese Tabellen gruppieren Funktionen und Fehlerbehebungen anhand der Quell-Engine.

Themen

- [Version 1.0.640 Oracle-Änderungen](#)
- [Version 1.0.640 Änderungen an Microsoft SQL Server](#)
- [Änderungen an MySQL-Version 1.0.640](#)
- [Version 1.0.640 PostgreSQL-Änderungen](#)
- [Änderungen an der Db2-LUW-Version 1.0.640](#)
- [Teradata-Änderungen an Version 1.0.640](#)
- [Version 1.0.640 ändert sich für andere Engines](#)

Version 1.0.640 Oracle-Änderungen

In der folgenden Tabelle sind Änderungen an Build 1.0.640 aufgelistet, in denen Oracle die Quell-Engine ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL L Aurora PostgreSQL L	Die Konvertierung von SQL-Code in Java- und Pro*C-Anwendungen wurde implementiert.
Oracle	PostgreSQL L Aurora PostgreSQL L	Die Leistung der folgenden Funktionen bei Verwendung in einer WHERE-Klausel wurde verbessert: <ul style="list-style-type: none"> • aws_oracle_ext.to_date • aws_oracle_ext.to_char • aws_oracle_ext.to_number • aws_oracle_ext.sysdate • aws_oracle_ext.sys_context
Oracle	RDS MariaDB 10.4	Zusätzliche Unterstützung für RDS MariaDB 10.4 für alle OLTP-Anbieter.
Oracle	PostgreSQL L/Aurora PostgreSQL L	Unterstützung für DBMS_UTILITY.GET_TIME wurde hinzugefügt. Folgende Emulationen wurden hinzugefügt: <ul style="list-style-type: none"> • DBMS_UTILITY.GET_TIME • DBMS_UTILITY.FORMAT_CALL_STACK • DBMS_UTILITY.CURRENT_INSTANCE
Oracle	MariaDB/M ySQL/Auro ra MySQL/ Microsoft SQL Server-Mo	Unterstützung der Freigabeklausel für TABLE(DATA,EXTENDED DATA), VIEW(DATA,EXTENDED DATA) und SEQUENCE(DATA) wurde hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
	dus=Orac e/Postgre SQL/ Aurora PostgreSQ L/RDS Oracle	
Oracle	PostgreSQ L/Aurora PostgreSQ L/Oracle RDS	<p>Die DEFAULT-Definition einer Spalte kann so erweitert werden, dass DEFAULT für die explizite NULL-Einfügung angewendet wird.</p> <p>Die DEFAULT-Klausel hat eine neue ON NULL-Klausel. Diese neue Klausel weist die Datenbank an, einen angegebenen Standardspaltenwert zuzuweisen, wenn eine INSERT-Anweisung versucht, einen Wert zuzuweisen, der als NULL ausgewertet wird.</p>
Oracle	MariaDB/ MariaDB (SQL MODE=ORA LE)	Zusätzliche Unterstützung für „Identitätsspalten“, die zum Zeitpunkt des Einfügens automatisch erhöht werden.
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	MariaDB 10.2/MariaDB 10.3/ MySQL/ Aurora MySQL/ PostgreSQL/ Aurora PostgreSQL	Die DEFAULT-Klausel verfügt über eine neue ON NULL-Klausel, die die Datenbank anweist, einen angegebenen Standardspaltenwert zuzuweisen, wenn eine INSERT-Anweisung versucht, einen Wert zuzuweisen, der als NULL ausgewertet wird.
Oracle	Oracle RDS/ MySQL /Aurora MySQL/ PostgreSQL/ Aurora PostgreSQL	Unterstützung für IDENTITY wurde hinzugefügt.
Oracle	MySQL 8.x	Unterstützung für CHECK-Einschränkung wurde hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Die Überprüfung von ANYDATA IS NULL/IS NOT NULL mithilfe der Erweiterungspack-Routine wurde implementiert.</p> <p>Die Emulation der VALUE-Funktion wurde implementiert, die in einer Abfrage verwendet wird, die auf der TABLE-Funktion von XMLSequence basiert.</p> <p>DBMS_LOB-Unterstützung für die folgenden integrierten Routinen wurde hinzugefügt:</p> <ul style="list-style-type: none"> • DBMS_LOB.CREATETEMPORARY • DBMS_LOB.FREETEMPORARY • DBMS_LOB.APPEND
Alle	SQL Server	<p>SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt.</p> <p>SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt.</p> <p>SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.</p>
Alle	Alle	<p>Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.</p>

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	Amazon Redshift	<p>Konvertierung von Cursorattributen in verschachtelten Blöcken wurde implementiert.</p> <p>Amazon Redshift unterstützt keine Sammlungen. Verwandte Variablen werden als VARCHAR konvertiert. Alle Erfassungsvorgänge außer dem Zuweisen einer Variablen zu einer anderen werden abgelehnt, einschließlich des Zugriffs auf Initiations- und Prüfelemente.</p> <p>Amazon Redshift Redshift-Verteilungsstil = AUTO implementiert.</p>

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Wenn ein nicht reserviertes Wort in Oracle in PostgreSQL reserviert ist, gilt Folgendes:</p> <ul style="list-style-type: none"> • Wenn das Wort in Anführungszeichen gesetzt wird, wird seine Groß- und Kleinschreibung beibehalten. • Wenn das Wort nicht in Anführungszeichen gesetzt wird, wird es in Großbuchstaben umgewandelt und in Anführungszeichen gesetzt. <p>Die Fähigkeit wird implementiert, Funktionen als Eingabe für LTRIM-, RTRIM- und TRIM-Funktionen zu verwenden.</p> <p>SELECT DISTINCT, ORDER BY-Ausdrücke müssen in der Auswahlliste angezeigt werden.</p> <p>Fügt für Cursorparameter, die auf einen Parameter mit einem DEFAULT-Wert folgen, die DEFAULT IS NULL-Klausel AWS SCT hinzu</p> <p>Source OUT-Cursorparameter werden in IN-Cursorparameter konvertiert.</p> <p>Die Paketvariable wird durch Hinzufügen der Option „Paketvariablen Logik-Implementierung“ unter „Konvertierungseinstellungen“ erneut implementiert. Verfügbare Einstellungen sind: „Sitzungsvariablen“ und „plv8 globale Objekte“. Der Standardwert ist „Sitzungsvariablen“.</p> <p>AUTONOMOUS_TRANSACTION-Pragma-Unterstützung mit dblink und pg_background wurde implementiert.</p>
Oracle	Alle	Ansicht SYS_%_TAB_COMMENTS wurde implementiert.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Oracle	PostgreSQL	Variableneingaben für Filter werden in PostgreSQL nicht unterstützt. Wenn bei der Konvertierung von Oracle nach PostgreSQL ein Variablenfilter festgestellt wird, wird nun eine Ausnahme gemeldet.
Oracle	Amazon Redshift	<p>Gespeicherter Code FOR..LOOP und Verbesserungen an Cursorkonvertierung wurden implementiert.</p> <p>Gespeicherter Codeaufruf von Funktionen/Prozeduren mit Standardparametern wurde implementiert.</p> <p>Gespeicherte Code-Fähigkeit zum UPDATE mit Alias ohne WHERE-Klausel wurde implementiert.</p> <p>Gespeicherte Codefunktionen, die zusätzliche Fälle mit SELECT FROM dual bilden, wurden implementiert.</p> <p>Gespeicherter Code Table%ROWTYPE-Parameter und Paketvariablen wurden implementiert.</p> <p>Gespeicherter Code, der von JAVA und externen Prozeduren verwendet wird, wurde implementiert.</p> <p>Standard-Oracle-Paket wurde in gespeichertem Code implementiert.</p>

Version 1.0.640 Änderungen an Microsoft SQL Server

Die folgende Tabelle listet Änderungen an Build 1.0.640 auf, in denen Microsoft SQL Server die Quell-Engine ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Microsoft Azure/Microsoft SQL Server	PostgreSQL/Aurora PostgreSQL/MySQL/	Unterstützung für COLUMN STORE-Indizes wurde hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
	Aurora MySQL	
Microsoft SQL Server	RDS MariaDB 10.4	Zusätzliche Unterstützung für RDS MariaDB 10.4 für alle OLTP-Anbieter.
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Unterstützung für das Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt.
Azure/SQL Server	MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Unterstützung für Datenbankknoten und Edgetabellentypen wurde hinzugefügt.
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Unterstützung für TEMPORAL TABLES wurde hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.
Azure/SQL Server	MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL/MariaDB	Unterstützung für die DML-Verarbeitung für SQL Server Graph Architecture hinzugefügt.
SQL Server	Aurora PostgreSQL	Die Option, Parameter ohne par_-Präfix zu konvertieren, wurde hinzugefügt.
Azure/SQL Server	MySQL 8.x	Unterstützung für CHECK-Einschränkung wurde hinzugefügt.
Alle	SQL Server	<p>SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt.</p> <p>SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt.</p> <p>SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.</p>
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SQL Server	AWS Glue (Python-SHELL)	<p>Verbesserungen an der Konvertierung, einschließlich:</p> <ul style="list-style-type: none"> • Integrierte Funktionskonvertierungen wurden in Python.String implementiert. • EXECUTE und EXEC wurden in gespeicherten Code implementiert. • Wurde mithilfe von Tabellentypen implementiert.
Azure/SQL Server	PostgreSQL/Aurora PostgreSQL	Wurde implementiert und macht \$TMP-Prozeduren optional.
SQL Server	MySQL/Aurora MySQL	<p>Erweiterte arithmetische Operationen mit Datumsangaben.</p> <p>Konstruktionsemulation 'TOP (Ausdruck) WITH TIES.</p> <p>Nach dem Aufruf von Prozeduren mit dem generierten Refcursor wird der Refcursor nun geschlossen.</p> <p>Das Festlegen einer GLOBAL-Isolationsebene wird in Aurora MySQL nicht unterstützt. Nur der Sitzungsbereich kann geändert werden. Das Standardverhalten von Transaktionen besteht darin, REPEATABLE READ und konsistente Lesevorgänge zu verwenden. Anwendungen, die für die Verwendung mit READ COMMITTED entwickelt wurden, müssen möglicherweise geändert werden. Alternativ können sie den Standardwert explizit in READ COMMITTED ändern.</p>

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SQL Server	AWS Glue (Python-SHELL)	<p>SQL Server-Anweisungen generieren eine vollständige Ergebnismenge, aber es gibt Zeiten, in denen die Ergebnisse am besten Zeile für Zeile verarbeitet werden. Das Öffnen eines Cursors in einer Ergebnismenge ermöglicht die zeilenweise Verarbeitung der Ergebnismenge. Sie können einen Cursor einer Variablen oder einem Parameter mit einem Cursor datentyp zuweisen.</p> <p>Implementiert das Einschließen einer Reihe von Transact-SQL-Anweisungen für gespeicherten Code, sodass eine Gruppe von Transact-SQL-Anweisungen ausgeführt werden kann, obwohl Python BEGIN und END von SQL Server nicht unterstützt. control-of-flow</p> <p>Die SQL Server-Anweisungen LABEL und GOTO werden von nicht unterstützt. AWS Glue Wenn AWS SCT ein Label im Code findet, wird es übersprungen. Wenn AWS SCT auf eine GOTO-Anweisung stößt, wird sie kommentiert.</p>
SQL Server	Amazon Redshift	<p>Die bedingte Verarbeitung von Transact-SQL-Anweisungen für den gespeicherten Code durch Implementierung des IF... ELSE-Steuer elements wurde implementiert.</p> <p>Implementiert, das eine Reihe von Transact-SQL-Anweisungen für gespeicherten Code einschließt, sodass eine Gruppe von Transact-SQL-Anweisungen als Block ausgeführt werden kann. Unterstützt verschachtelte BEGIN... END-Blöcke.</p> <p>SET und SELECT wurden in gespeicherten Code implementiert.</p> <p>CREATE INDEX wurde in Amazon Redshift (das keine Indizes unterstützt) implementiert, indem ein benutzerdefinierter Sortierschlüssel für die Tabellen erstellt wurde.</p>

Änderungen an MySQL-Version 1.0.640

In der folgenden Tabelle sind Änderungen an Build 1.0.640 aufgelistet, in denen MySQL die Quell-Engine ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
MySQL	PostgreSQL 12.x	Unterstützung für generierte Spalten wurde hinzugefügt.
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.
MySQL	PostgreSQL/Aurora PostgreSQL 11.	Unterstützung für Folgendes wurde hinzugefügt: <ul style="list-style-type: none"> • Transaktionen wurden in gespeicherte SQL-Prozeduren eingebettet. • Die Möglichkeit, gespeicherte SQL-Prozeduren aufzurufen. • Die Möglichkeit, gespeicherte SQL-Prozeduren zu erstellen.
Alle	SQL Server	<p>SQL Server 2019: Unterstützung für das neue Indexattribut <code>OPTIMIZE_FOR_SEQUENTIAL_KEY</code> wurde hinzugefügt.</p> <p>SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt.</p> <p>SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.</p>
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.

Version 1.0.640 PostgreSQL-Änderungen

In der folgenden Tabelle sind Änderungen an Build 1.0.640 aufgelistet, in denen PostgreSQL die Quell-Engine ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
PostgreSQL	MySQL 8.x	<p>MySQL unterstützt jetzt die Erstellung von funktionalen Indexschlüsselteilen, die Ausdruckswerte und nicht Spaltenwerte indizieren. Funktionale Schlüsselteile ermöglichen die Indizierung von Werten, die nicht auf andere Weise indiziert werden können, z. B. JSON-Werte.</p> <p>MySQL unterstützt jetzt CTE und rekursive CTE.</p>
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.
PostgreSQL 11.x	PostgreSQL/Aurora PostgreSQL 11.	<p>Unterstützung für Folgendes wurde hinzugefügt:</p> <ul style="list-style-type: none"> • Transaktionen wurden in gespeicherte SQL-Prozeduren eingebettet. • Die Möglichkeit, gespeicherte SQL-Prozeduren aufzurufen. • Die Möglichkeit, gespeicherte SQL-Prozeduren zu erstellen.
PostgreSQL	MySQL 8.x	<p>MySQL-Unterstützung für absteigende Indizes wurde hinzugefügt. DESC in einer Indexdefinition wird nicht mehr ignoriert, sondern bewirkt, dass Schlüsselwerte in absteigender Reihenfolge gespeichert werden.</p> <p>MySQL-Unterstützung wurde hinzugefügt zur Verwendung von Ausdrücken als Standardwerte in Datentypspezifikationen, einschlie</p>

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		<p data-bbox="521 212 1487 289">ßlich Ausdrücke als Standardwerte für die Datentypen BLOB, TEXT, GEOMETRY und JSON.</p> <p data-bbox="521 338 1393 415">Mehrere vorhandene Aggregationsfunktionen können jetzt als Fensterfunktionen verwendet werden.</p> <ul data-bbox="521 464 860 1423" style="list-style-type: none">• AVG()• BIT_AND()• BIT_OR()• BIT_XOR()• COUNT()• JSON_ARRAYAGG()• JSON_OBJECTAGG()• MAX()• MIN()• STDDEV_POP()• STDDEV()• STD()• STDDEV_SAMP()• SUM()• VAR_POP()• VARIANCE()• VAR_SAMP() <p data-bbox="521 1528 1446 1661">MySQL unterstützt Fensterfunktionen, die für jede Zeile aus einer Abfrage eine Berechnung unter Verwendung zu dieser Zeile gehöriger Zeilen durchführen</p> <ul data-bbox="521 1703 803 1856" style="list-style-type: none">• CUME_DIST()• DENSE_RANK()• FIRST_VALUE()

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		<ul style="list-style-type: none"> • LAG() • LAST_VALUE() • LEAD() • NTH_VALUE() • NTILE() • PERCENT_RANK() • RANK() • ROW_NUMBER()
PostgreSQL	MySQL 8.x	Unterstützung für CHECK-Einschränkung wurde hinzugefügt.
Alle	SQL Server	<p>SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt.</p> <p>SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt.</p> <p>SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.</p>
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.
PostgreSQL/Aurora PostgreSQL	Alle	<p>Systemansicht mit Sysindexes-Emulation wurde hinzugefügt.</p> <p>Wenn eine SELECT-Anweisung in einer Prozedur ohne Angabe von INTO vorhanden ist, wird der Parameter INOUT p_refcur vom Typ refcursor für eine Prozedur am Ziel erstellt.</p>

Änderungen an der Db2-LUW-Version 1.0.640

In der folgenden Tabelle sind Änderungen an Build 1.0.640 aufgelistet, in denen DB2 LUW die Quell-Engine ist.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
DB2 LUW	RDS MariaDB 10.4	Zusätzliche Unterstützung für RDS MariaDB 10.4 für alle OLTP-Anbieter.
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.
DB2 LUW	MySQL 8.0.17	Unterstützung für CHECK-Beschränkungen wurde hinzugefügt.
Alle	SQL Server	<p>SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt.</p> <p>SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt.</p> <p>SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.</p>
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.

Teradata-Änderungen an Version 1.0.640

In der folgenden Tabelle sind die Änderungen an den Teradata-Quellmodulen in Build 1.0.640 aufgeführt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Teradata	Amazon Redshift	<p>Unterstützung für die Anweisungen MERGE und QUALIFY wurde hinzugefügt.</p> <p>LOCKING ROWS FOR ACCESS-Klausel wurde von Teradata-Anweisungen entfernt.</p> <p>Unterstützung für CAST-Funktion wurde hinzugefügt.</p>
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Teradata	Teradata	Verbesserungen wurden in REGEXP_INSTR() and REGEXP_SUBSTR() implementiert.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.
Alle	SQL Server	<p>SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt.</p> <p>SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt.</p> <p>SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.</p>
Teradata	Alle	Unterstützung wurde für REGEXP_INSTR() und REGEXP_SUBSTR() hinzugefügt.
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.
Teradata	Amazon Redshift	Die Fähigkeit wurde implementiert, den SQL-Code der Quellstruktur je nach Phase in eine einzelne Datei oder mehrere Dateien mit den

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
		Einstellungen in den „Projekteinstellungen“, „Als SQL speichern“ und „Übernehmen“, „Dropdown-Liste“, „Einzelne Datei/Mehrere Dateien“ zu speichern. Verbesserungen bei Ansichten und Prozedurkonvertierungen wurden implementiert.
Teradata	Alle	Unterstützung für Teradata-Version 16.20 wurde hinzugefügt

Version 1.0.640 ändert sich für andere Engines

Die folgende Tabelle listet Änderungen an Build 1.0.640 für andere Quell-Engines auf.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
Sybase	RDS MariaDB 10.4	Zusätzliche Unterstützung für RDS MariaDB 10.4 für alle OLTP-Anbieter.
SAP ASE	MariaDB	Folgendes wurde implementiert: <ul style="list-style-type: none"> • MariaDB 10.4 • EXECUTE IMMEDIATE-Anweisung • DEFAULT-Definitionen • Unterstützung der CHECK-Beschränkungen
SAP ASE	PostgreSQL 12.x	Unterstützung für generierte Spalten wurde hinzugefügt.
Alle	Alle	Aktualisieren Sie von JDK 8 auf Amazon Corretto JDK 11. Weitere Informationen, einschließlich Download-Links, finden Sie unter Was ist Amazon Corretto 11? im Amazon Corretto 11-Benutzerhandbuch.
Alle	Alle	Dem Bewertungsbericht wurden Informationen über mögliche Inkonsistenzen in der Datenbank des Benutzers hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SAP ASE	MySQL 8.0.17	Unterstützung für CHECK-Beschränkungen wurde hinzugefügt.
Alle	SQL Server	<p>SQL Server 2019: Unterstützung für das neue Indexattribut OPTIMIZE_FOR_SEQUENTIAL_KEY wurde hinzugefügt.</p> <p>SQL Server 2017: Unterstützung für Diagrammdatenbanken-Knoten- und Edgetabellentypen wurde hinzugefügt.</p> <p>SQL Server 2016: Unterstützung für TEMPORAL TABLES hinzugefügt.</p>
Vertica	Amazon Redshift	Unterstützung für den Verteilungsstil = AUTO wurde hinzugefügt.
Alle	Alle	Es wurde die Möglichkeit implementiert, physische Partitionen durch virtuelle Partitionen zu überschreiben. Data Warehouse-Extraktoren extrahieren Daten gemäß erstellten virtuellen Partitionen.
Amazon Redshift	Amazon Redshift	Nicht unterstützte integrierte Funktionen in DML-Anweisungen werden durch NULL als Platzhalter ersetzt.
Sybase	PostgreSQL	Unterstützung für native Funktionen hinzugefügt.
SAP ASE	MySQL/ Aurora MySQL	Die standardmäßige Isolationsebene für Aurora MySQL ist REPEATABLE READ. Das Festlegen einer GLOBAL-Isolationsebene wird in Aurora MySQL nicht unterstützt. Nur der Sitzungsbereich kann geändert werden. Das Standardverhalten von Transaktionen besteht darin, REPEATABLE READ und konsistente Lesevorgänge zu verwenden. Anwendungen, die mit READ COMMITTED ausgeführt werden sollen, müssen möglicherweise geändert werden. Sie können auch den Standardwert explizit in READ COMMITTED ändern.
SAP ASE	PostgreSQL	Unterstützung für die CONVERT-Funktion (optimistisch) ohne das Erweiterungspaket hinzugefügt.

Quelle	Ziel	Neuerungen, Verbesserungen oder Korrekturen
SAP ASE	Alle	<p>Systemansicht mit Sysindexes-Emulation wurde hinzugefügt.</p> <p>Wenn eine SELECT-Anweisung in einer Prozedur ohne Angabe von INTO vorhanden ist, wird der Parameter INOUT p_refcur vom Typ refcursor für eine Prozedur am Ziel erstellt.</p>
Greenplum	Amazon Redshift	CREATE TEMORATE TABLE wurde wie folgt implementiert:

Dokumentverlauf

In der folgenden Tabelle werden die wichtigen Änderungen am AWS Schema Conversion Tool (AWS SCT) -Benutzerhandbuch nach Januar 2018 beschrieben.

Sie können einen RSS-Feed abonnieren, um Benachrichtigungen über Aktualisierungen dieser Dokumentation zu erhalten.

Änderung	Beschreibung	Datum
AWS SCTbuild #1 .0.672	Build 1.0.672 unterstützt Amazon RDS für PostgreSQL 15 als Ziel und Microsoft SQL Server Version 2022 als Quelle. Es bietet auch Unterstützung für neue Amazon Redshift-Funktionen im konvertierten Code, implementiert mehrere Konvertierungsverbesserungen für IBM Db2 for z/OS Source und behebt eine Reihe von Konvertierungsproblemen.	8. Mai 2023
AWS SCTbuild #1 .0.671	Build 1.0.671 unterstützt Migrationen von Apache Oozie nach AWS Step Functions. Es bietet auch Unterstützung für BigQuery als Quelle für den Multiserver-Bewertungsprozess. Darüber hinaus fügt es neue Konvertierungseinstellungen für IBM Db2 for z/OS als Quelle hinzu und behebt eine Reihe von Konvertierungsproblemen.	08. März 2023

[AWS SCTbuild #1 .0.670](#)

Build 1.0.670 unterstützt Migrationen von Hadoop zu Amazon EMR. Es bietet auch Unterstützung für Azure Synapse Analytics als Quelle für den Multiserver-Bewertungsprozess. Darüber hinaus verbessert es die Konvertierung von SQL-Code, der in Java-Anwendungen eingebettet ist, und behebt eine Reihe von Konvertierungsproblemen.

23. Januar 2023

[AWS SCTbuild #1 .0.669](#)

Build 1.0.669 implementiert die Unterstützung der nativen Partitionierung für die Datenmigration aus Oracle Data Warehouse. Es verbessert auch den Multiserver-Bewertungsprozess, fügt neue Funktionen für Datenextraktions-Agents hinzu und behebt eine Reihe von Konvertierungsproblemen.

19. Dezember 2022

[AWS SCTbuild #1 .0.668](#)

Build 1.0.668 implementiert die automatische virtuelle Partitionierung für die Datenmigration aus Greenplum-Datenbanken und bietet Unterstützung für die Datenmigration von Snowflake-Datenbanken zu Amazon Redshift. Es verbessert auch die Konvertierung von SQL-Code, der in C#-Anwendungen eingebettet ist, und behebt eine Reihe von Konvertierungsproblemen.

16. November 2022

[AWS SCTbuild #1 .0.667](#)

Build 1.0.667 bietet Unterstützung für die ETL-Engine (Extract, Transform, Load) von Informatica als Migrationssquelle. Es aktualisiert auch die Version des Erweiterungspakets, erhöht die unterstützte Mindesttreiberversion für Amazon Redshift und behebt eine Reihe von Konvertierungsproblemen.

13. Oktober 2022

[AWS SCTbuild #1 .0.666](#)

Build 1.0.666 verbessert die Konvertierung von Java-Anwendungen, indem Unterstützung für das MyBatis Framework hinzugefügt wird. Es fügt Erweiterungspaketen auch neue Funktionen hinzu, verbessert den Quell-Metadaten-Loader und behebt eine Reihe von Konvertierungsproblemen.

20. September 2022

[AWS SCTbuild #1 .0.665](#)

Build 1.0.665 bietet Unterstützung BigQuery als Migration squelle. Es implementiert auch die Unterstützung der neuen Version der Konfigurationsdatei für Babelfish-Funktionen. Darüber hinaus verbessert es die Konvertierung von Data Warehouses zu Amazon Redshift und behebt eine Reihe von Konvertierungsproblemen.

20. August 2022

[AWS SCTbuild #1 .0.664](#)

Build 1.0.664 unterstützt Amazon Redshift Serverless als Migrationsquelle oder -ziel. Es implementiert auch einen automatischen Speicherausgleich bei Datenextraktionsaufgaben und behebt einen Fehler, bei dem keine Verbindung zu AWS Snowball Geräten hergestellt werden konnte. Darüber hinaus bietet es die Möglichkeit, die Spaltensortierung in den Migrationsregeln zu ändern, verbessert die Benutzeroberfläche und behebt eine Reihe von Konvertierungsproblemen.

14. Juli 2022

[AWS SCTbuild #1 .0.663](#)

Build 1.0.663 bietet Unterstützung für Babelfish for Aurora PostgreSQL 1.2.0 und verbessert die Funktionen von Bewertungsberichten auf mehreren Servern. Es fügt auch neue Funktionen in die Migrationsregeln ein, behebt zwei Loader-Fehler und behebt eine Reihe von Konvertierungsproblemen.

20. Juni 2022

[AWS SCTbuild #1 .0.662](#)

Build 1.0.662 implementiert die SQL-Codekonvertierung in C#-Anwendungen und verbessert den Arbeitsablauf für Bewertungsberichte auf mehreren Servern. Es enthält auch mehrere Konvertierungsverbesserungen und behebt eine Reihe von Konvertierungsproblemen.

19. Mai 2022

[AWS SCTbuild #1 .0.661](#)

Build 1.0.661 unterstützt IBM Db2 for z/OS als Migrationsquelle. Es bietet auch Unterstützung für die Konvertierung von ETL-Skripten (Extract, Transform, Load) in AWS Glue Studio und behebt eine Reihe von Konvertierungsproblemen.

21. April 2022

[AWS SCTbuild #1 .0.660](#)

Build 1.0.660 unterstützt die PostgreSQL-Hauptversion 14 und MariaDB 10.6 als Migrationsziele. Es bietet auch Unterstützung für die Konvertierung von Oracle-Indizes für materialisierte Ansichten und behebt eine Reihe von Konvertierungsproblemen.

21. März 2022

[AWS SCTbuild #1 .0.659](#)

Build 1.0.659 unterstützt die PostgreSQL-Hauptversion 13 auf der Aurora PostgreSQL-kompatiblen Edition als Migrationsziel. Es implementiert die SQL-Code-Konvertierung in C#-Anwendungen, bietet Unterstützung für Oracle Unified Auditing und behebt eine Reihe von Konvertierungsproblemen.

21. Februar 2022

[AWS SCTbuild #1 .0.658](#)

Build 1.0.658 bietet die Integration mit der virtuellen Zieldatenbankplattform Amazon Redshift AWS Secrets Manager und fügt Unterstützung hinzu. Es enthält auch eine Reihe von Konvertierungsverbesserungen und Bugfixes.

20. Januar 2022

[AWS SCTbuild #1 .0.657](#)

Build 1.0.657 verbessert die Konvertierung von Microsoft SQL Server zu Aurora PostgreSQL-Compatible Edition, Amazon RDS for PostgreSQL und anderen Migrationszielen. Es enthält auch eine Reihe von Verbesserungen der Benutzeroberfläche und Bugfixes.

20. Dezember 2021

[AWS SCTbuild #1 .0.656](#)

Build 1.0.656 unterstützt mehrere Quell- und Zieldatenbanken in einem Projekt. Es enthält auch Konvertierungs-, Optimierungsstrategien und allgemeine Verbesserungen sowie eine Reihe von Bugfixes.

22. November 2021

[AWS SCTbuild #1 .0.655](#)

Build 1.0.655 implementiert die Konvertierung von FastExport Teradata-Jobskripten zu Amazon Redshift RSQL und erhöht die unterstützte Mindesttreiberversion für Greenplum auf 42.2.5. Es enthält auch eine Reihe von Verbesserungen und Bugfixes.

18. Oktober 2021

[AWS SCTbuild #1 .0.654](#)

Build 1.0.654 implementiert die Konvertierung von Shell-, Teradata- und Teradata Basic Teradata FastLoad Query (BTEQ) -Skripten zu Amazon Redshift RSQL. Es behebt auch eine Reihe von Konvertierungsproblemen und fügt eine Reihe von Verbesserungen und Bugfixes hinzu.

16. September 2021

[AWS SCTbuild #1 .0.653](#)

Build 1.0.653 implementiert die Konvertierung von dynamischem SQL, das in aufgerufenen Funktionen oder Prozeduren erstellt wurde. Es verbessert auch die Konvertierung verschlüsselter Routinen und fügt eine Reihe von Verbesserungen und Bugfixes hinzu.

10. August 2021

[AWS SCTbuild #1 .0.652](#)

Build 1.0.652 implementiert den Skriptbefehlsmodus in der Befehlszeilenschnittstelle und implementiert Regeln zur Schemaoptimierung. Es enthält auch eine Reihe von Konvertierungs- und Leistungsverbesserungen sowie Bugfixes.

30. Juni 2021

[AWS SCTbuild #1 .0.651](#)

Build 1.0.651 enthält eine Reihe von Verbesserungen und Bugfixes. Es bietet auch Zugriff auf die erste Kopie der AWS Schema Conversion ToolCLI-Referenz.

4. Juni 2021

[AWS SCTbuild #1 .0.650](#)

Build 1.0.650 implementiert die Unterstützung von Amazon RDS für PostgreSQL 13 als Zieldatenbank und aktualisiert Extraktor-Agenten. Es aktualisiert auch die unterstützte Mindestversion des JDBC-Treibers für Microsoft SQL Server, Azure und Azure Synapse. Darüber hinaus enthält es eine Reihe von Konvertierungsverbesserungen und Bugfixes.

30. April 2021

[AWS SCTbuild #1 .0.649](#)

Build 1.0.649 implementiert die Unterstützung von MariaDB 10.5 als Zieldatenbank und implementiert Funktionserweiterungen für die Konvertierung der in Oracle integrierten Funktionen. Es enthält auch eine Reihe von Konvertierungs- und Leistungsverbesserungen sowie Bugfixes.

29. März 2021

[AWS SCTbuild #1 .0.648](#)

Build 1.0.648 fügt eine Reihe von Konvertierungsverbesserungen und Bugfixes hinzu.

22. Februar 2021

AWS SCTbuild #1 .0.647	Build 1.0.647 bietet Unterstützung für die Funktion Database Mail in Amazon RDS und implementiert das Laden und Konvertieren von Kommentaren zu Speicherobjekten. Außerdem AWS SCT werden Data Migration Service Assessor und AWS SCT Wizard hinzugefügt und die Baumfilter-Benutzeroberfläche implementiert. Darüber hinaus enthält es einen neu gestalteten Abschnitt im Bewertungsbericht sowie eine Reihe von Verbesserungen und Bugfixes.	15. Januar 2021
AWS SCTbuild #1 .0.646	Build 1.0.646 bietet Unterstützung für INTERVAL-Datentypen, Identitätsspalten und Cursorkonvertierung und fügt eine Reihe von Verbesserungen und Bugfixes hinzu.	28. Dezember 2020
AWS SCTbuild #1 .0.645	Build 1.0.645 fügt Unterstützung für ETL SSIS zur AWS Glue Konvertierung sowie eine Reihe von Verbesserungen und Bugfixes hinzu.	16. November 2020
AWS SCTbuild #1 .0.643-1.0.644	Build 1.0.644 enthält eine Reihe von Konvertierungs-, Leistungs- und Benutzeroberflächenverbesserungen sowie Bugfixes.	14. Oktober 2020

<u>AWS SCTbuild #1 .0.642</u>	Build 1.0.642 implementiert die Konvertierung von ETL-Paketen von Microsoft SQL Server Integration Services zu AWS Glue und fügt eine Reihe von Verbesserungen und Bugfixes hinzu.	28. August 2020
<u>AWS SCTbuild #1 .0.641</u>	SSL-Unterstützung für Datenextraktoren hinzugefügt. Build enthält auch eine Reihe von Verbesserungen und Korrekturen.	17. Juli 2020
<u>AWS SCTBuilds #1 .0.633-1.0.640</u>	Aktualisierung von JDK 8 auf Amazon Corretto JDK 11. Hinzufügung von Tabellen, die andere Aktualisierungen, Änderungen und Korrekturen identifizieren.	22. Juni 2020
<u>Verfügbarkeit von AWS WQF</u>	AWS SCT stellt das AWS Workload Qualification Framework (AWS WQF)-Tool nicht mehr zum Download bereit.	19. Juni 2020

[AWS SCTBuilds #1 .0.632](#)

SCT UI - Neue Registerkarte hinzugefügt, um Fehler anzuzeigen, die beim Anwenden von Skripten auftreten. Sie können die Quellstruktur nun als SQL speichern, wenn Sie aus SAP ASE konvertieren. Verbesserungen für Konvertierungen in PostgreSQL oder Aurora PostgreSQL oder Redshift.

19. November 2019

[AWS SCTBuilds #1 .0.631 und #1 .0.630 \(kombiniert\)](#)

Bessere Unterstützung von RowIDs in Oracle und für Systemobjekte in Microsoft SQL Server und SAP ASE. Bessere Handhabung fehlender Spezifizierer von SQL Server-Schemas. Bessere Unterstützung für Konvertierungen von Greenplum in Redshift. Verbesserte Unterstützung für die Konvertierung von gespeichertem Code bei der Umstellung auf Amazon Redshift, MariaDB, MySQL und PostgreSQL.

30. September 2019

AWS SCTbuild #1 .0.629	Unterstützung für gespeicherte Prozeduren für Konvertierungen von Netezza. Verbesserte Unterstützung für Konvertierungen zu Amazon Redshift, DynamoDB, MySQL und PostgreSQL. Es wurde Unterstützung für SAP ASE 12.5 als Quelle hinzugefügt.	20. August 2019
AWS SCTbuild #1 .0.628	Unterstützung der Service-Emulation für Konvertierungen von DB2, SQL Server und Oracle. Verbesserungen für Konvertierungen zu Amazon Redshift, einschließlich mehr Unterstützung für Cursor und gespeicherte Prozeduren.	22. Juni 2019
AWS SCTbuild #1 .0.627	Unterstützung für Konvertierungen von SQL Server zu gespeicherten Prozeduren in Amazon Redshift. Verbesserungen für die Konvertierung zu PostgreSQL 11 und MySQL 8.0.	31. Mai 2019
AWS SCTbuild #1 .0.626	PostgreSQL 11 und MySQL 8.0 werden jetzt als Ziele unterstützt. SAP ASE 15.5 wird nun als Quelle unterstützt.	26. April 2019

[AWS SCTbuild #1 .0.625](#)

Zu den Updates gehören die Möglichkeit, Teradata BTEQ zu konvertieren AWS Glue, Unterstützung für Konvertierungen zu MariaDB 10.3 mit Unterstützung für den Oracle-Kompatibilitätsmodus , Unterstützung für SAP ASE 15.7 und Service-Substitutionen zur Emulation fehlender Funktionen.

25. März 2019

[AWS SCTbuild #1 .0.624](#)

Zu den Updates gehören die Möglichkeit, Oracle ETL zu konvertieren AWS Glue, und die Unterstützung von Konvertierungen von Microsoft SQL Server, Oracle und IBM Db2 LUW zu Amazon RDS for MariaDB. Wir haben auch Unterstützung für die Konvertierungen von SAP ASE in RDS für MySQL sowie Amazon Aurora mit MySQL-Kompatibilität hinzugefügt. Darüber hinaus haben wir Unterstützung für die Oracle-Erweiterung Oracle bei der Oracle-Konvertierung in PostgreSQL hinzugefügt.

22. Februar 2019

AWS SCTbuild #1 .0.623	Die Updates bieten folgende Konvertierungsmöglichkeiten : SAP ASE-Datenbanken sowie die Konvertierung von T-SQL-Skripts, DML und DDL in entsprechenden Code oder Komponenten. Wir haben auch Oracle- und Microsoft SQL Server-Emulatoren für bessere Konvertierungen hinzugefügt.	25. Januar 2019
AWS SCTbuild #1 .0.622	Aktualisierungen umfassen das Workload Qualification Framework, das den Workload für eine gesamte Migration, einschließlich Datenbank- und App-Änderungen, analysiert.	20. Dezember 2018
AWS SCTbuild #1 .0.621	Aktualisierungen beinhalten Unterstützung für Aurora PostgreSQL 10 als Ziel und die Möglichkeit zur Migration von Netezza mit externen Tabellenoptionen.	21. November 2018
AWS SCTbuild #1 .0.620	Aktualisierungen beinhalten die Fähigkeit, SQL-Skripts zu speichern, sowie Unterstützung für globale Oracle-Cursor bei der Migration zu MySQL.	22. Oktober 2018
AWS SCTbuild #1 .0.619	Die Aktualisierung unterstützt unter anderem die Migration von Apache Cassandra in DynamoDB und Vertica 9 als Quelle.	20. September 2018

AWS SCTbuild #1 .0.618	Updates umfassen erweiterte Bewertungsberichte, Unterstützung der Konvertierung von Oracle ROWIDs und Unterstützung von benutzerdefinierten SQL Server-Tabellen.	24. August 2018
AWS SCTbuild #1 .0.617	Updates umfassen erweiterte Bewertungsberichte, Unterstützung der Konvertierung von Oracle ROWIDs und Unterstützung von benutzerdefinierten SQL Server-Tabellen.	24. Juli 2018
AWS SCTbuild #1 .0.616	Updates beinhalten Unterstützung von RDS bei der Konvertierung von Oracle in Amazon RDS for Oracle und bei der Konvertierung von Oracle-Zeitplanobjekten sowie Unterstützung von Oracle-Jobs, Partitionierung und Db2 LUW Version 10.1.	26. Juni 2018
AWS SCTbuild #1 .0.615	Updates beinhalten Unterstützung für SQL Server in PostgreSQL GOTO-Anweisungen, PostgreSQL 10 Partitionierung und Db2 LUW Version 10.1.	24. Mai 2018
AWS SCTbuild #1 .0.614	Updates beinhalten Unterstützung von Oracle zu Oracle DB Links, SQL Server in PostgreSQL-Inline-Funktionen und die Emulation von Oracle-Systemobjekten.	25. April 2018

[AWS SCTbuild #1 .0.613](#)

Updates beinhalten Unterstützung von Db2 LUW, die Konvertierung von SQL* Plus-Dateien und SQL Server Windows-Authentifizierung.

28. März 2018

[AWS SCTbuild #1 .0.612](#)

Updates beinhalten Unterstützung für benutzerdefinierte Datentypzuweisung, einen Schemavergleich für Oracle 10 und die Umwandlung globaler Variablen von Oracle zu PostgreSQL.

22. Februar 2018

[AWS SCTbuild #1 .0.611](#)

Updates beinhalten Unterstützung von dynamischen Anweisungen von Oracle zu PostgreSQL, das Öffnen der Protokolldatei durch Auswahl einer Fehlermeldung und die Möglichkeit, Schemas in einer Strukturansicht auszublenden.

23. Januar 2018

Frühere Aktualisierungen

In der folgenden Tabelle werden die wichtigen Änderungen am AWS Schema Conversion Tool (AWS SCT) -Benutzerhandbuch vor Januar 2018 beschrieben.

Version	Änderung	Beschreibung	Datum geändert
1.0.608	FIPS-Endpunktunterstützung für Amazon S3	Sie können jetzt mithilfe von FIPS-Endpunkten eine Verbindung AWS SCT zu Amazon S3 und Amazon Redshift beantragen, um die Sicherheitsanforderungen des Federal Information Processing Standard zu	17. November 2017

Version	Änderung	Beschreibung	Datum geändert
		erfüllen. Weitere Informationen finden Sie unter AWSAnmeldeinformationen speichern .	
1.0.607	FIPS-Endpunktunterstützung für Amazon S3	Sie können jetzt mithilfe von FIPS-Endpunkten eine Verbindung AWS SCT zu Amazon S3 und Amazon Redshift beantragen, um die Sicherheitsanforderungen des Federal Information Processing Standard zu erfüllen. Weitere Informationen finden Sie unter AWSAnmeldeinformationen speichern .	30. Oktober 2017
1.0.607	Datenextraktionsaufgaben können LOBs ignorieren	Wenn Sie Datenextraktionsaufgaben erstellen, können Sie jetzt große Objekte (LOBs) ignorieren, um die Datenmenge zu reduzieren, die Sie extrahieren. Weitere Informationen finden Sie unter Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen .	30. Oktober 2017
1.0.605	Zugriff auf das Aufgabenprotokoll des Agenten für die Datenextraktion	Sie können jetzt über einen benutzerfreundlichen Link auf der AWS Schema Conversion Tool-Benutzeroberfläche auf das Aufgabenprotokoll des Agenten für die Datenextraktion zugreifen. Weitere Informationen finden Sie unter Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen .	28. August 2017
1.0.604	Converter-Erweiterungen	Die AWS Schema Conversion Tool-Engine verfügt jetzt über eine verbesserte Konvertierung für heterogene Migrationen.	24. Juni 2017

Version	Änderung	Beschreibung	Datum geändert
1.0.603	Filter für den Agenten der Datenextraktion	Sie können jetzt die Daten filtern, die die Agenten für die Datenextraktion aus dem Data Warehouse extrahieren. Weitere Informationen finden Sie unter Regeln für die Datenmigration erstellen in AWS SCT .	16. Juni 2017
1.0.603	AWS SCT unterstützt zusätzliche Data Warehouse-Versionen	Sie können jetzt die verwenden AWS Schema Conversion Tool, um Ihre Teradata 13- und Oracle Data Warehouse 10-Schemas in entsprechende Amazon Redshift-Schemas zu konvertieren. Weitere Informationen finden Sie unter Konvertierung von Data Warehouse-Schemas in Amazon Redshift mithilfe von AWS SCT .	16. Juni 2017
1.0.602	Unterstützung zusätzlicher Data Warehouses durch die Agenten für die Datenextraktion	Sie können jetzt mit Agenten der Datenextraktion Daten aus Ihren Microsoft SQL Server Data Warehouses extrahieren. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift .	11. Mai 2017

Version	Änderung	Beschreibung	Datum geändert
1.0.602	Datenextraktionsagenten können Daten nach Amazon Redshift kopieren	Die Agenten für die Datenextraktion unterstützen jetzt drei Modi für das Hochladen. Sie können jetzt angeben, ob Sie nur Ihre Daten extrahieren, Ihre Daten extrahieren und einfach auf Amazon S3 hochladen oder Ihre Daten direkt in Amazon Redshift extrahieren, hochladen und kopieren möchten. Weitere Informationen finden Sie unter Eine AWS SCT Datenextraktionsaufgabe erstellen, ausführen und überwachen .	11. Mai 2017
1.0.601	AWS SCT unterstützt zusätzliche Data Warehouses	Sie können das jetzt verwenden AWS Schema Conversion Tool, um Ihre Vertica- und Microsoft SQL Server-Schemas in entsprechende Amazon Redshift-Schemas zu konvertieren. Weitere Informationen finden Sie unter Konvertierung von Data Warehouse-Schemas in Amazon Redshift mithilfe von AWS SCT .	18. April 2017
1.0.601	Unterstützung zusätzlicher Data Warehouses durch die Agenten für die Datenextraktion	Sie können jetzt mit Agenten der Datenextraktion Daten aus Ihren Greenplum-, Netezza- und Vertica-Data Warehouses extrahieren. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift .	18. April 2017

Version	Änderung	Beschreibung	Datum geändert
1.0.601	Unterstützung zusätzlicher Betriebssysteme durch die Agenten für die Datenextraktion	Sie können nun Agenten für die Datenextraktion auf Computern installieren, auf denen die Betriebssysteme Microsoft Windows oder macOS ausgeführt werden. Weitere Informationen finden Sie unter Installation von Extraktionsagenten .	18. April 2017
1.0.601	Agenten zur Datenextraktion laden automatisch auf Amazon S3 hoch	Datenextraktions-Agenten laden Ihre extrahierten Daten jetzt automatisch auf Amazon S3 hoch. Weitere Informationen finden Sie unter Ausgabe der Aufgabe zur Datenextraktion .	18. April 2017
1.0.600	Agenten für die Datenextraktion	Sie können jetzt Datenextraktionsagenten installieren, die Daten aus Ihrem Data Warehouse extrahieren und für die Verwendung mit Amazon Redshift vorbereiten. Sie können mit dem AWS Schema Conversion Tool die Agenten registrieren und Datenextraktionsaufgaben für diese erstellen. Weitere Informationen finden Sie unter Migrieren von Daten aus einem lokalen Data Warehouse zu Amazon Redshift .	16. Februar 2017

Version	Änderung	Beschreibung	Datum geändert
1.0.600	Kunden-Feedback	Sie können jetzt Feedback zum AWS Schema Conversion Tool bereitstellen. Sie können einen Fehlerbericht übermitteln, eine Funktionalitätsanforderung stellen oder allgemeine Informationen bereitstellen. Weitere Informationen finden Sie unter Feedback geben .	16. Februar 2017
1.0.502	Integration in AWS DMS	Sie können jetzt mit dem AWS Schema Conversion Tool AWS DMS-Endpunkte und -Aufgaben erstellen. Sie können die Aufgaben mit dem AWS SCT ausführen und überwachen. Weitere Informationen finden Sie unter Verwenden von AWS SCT mit AWS DMS .	20. Dezember 2016
1.0.502	Amazon Aurora mit PostgreSQL-Kompatibilität als Zieldatenbank	Das AWS Schema Conversion Tool unterstützt jetzt Amazon Aurora mit PostgreSQL-Kompatibilität als Zieldatenbank. Weitere Informationen finden Sie unter Konvertieren von Datenbankschemas mithilfe von AWS SCT .	20. Dezember 2016
1.0.502	Profilunterstützung	Sie können jetzt verschiedene Profile im AWS Schema Conversion Tool speichern und unkompliziert zwischen diesen wechseln. Weitere Informationen finden Sie unter Speichern von AWS Serviceprofilen in AWS SCT .	20. Dezember 2016

Version	Änderung	Beschreibung	Datum geändert
1.0.501	Support für Greenplum Database und Netezza	Sie können jetzt die verwenden, AWS Schema Conversion Tool um Ihre Data Warehouse-Schemas von Greenplum Database und Netezza nach Amazon Redshift zu konvertieren. Weitere Informationen finden Sie unter Konvertierung von Data Warehouse-Schemas in Amazon Redshift mithilfe von AWS SCT .	17. November 2016
1.0.501	Redshift-Optimierung	Sie können das jetzt verwendenAWS Schema Conversion Tool, um Ihre Amazon Redshift-Datenbanken zu optimieren. Weitere Informationen finden Sie unter Optimierung von Amazon Redshift mithilfe von AWS SCT .	17. November 2016
1.0.500	Zuweisen von Regeln	Bevor Sie das Schema mit dem AWS Schema Conversion Tool konvertieren, können Sie jetzt Regeln festlegen, um den Datentyp von Spalten zu ändern, Objekte von einem Schema in ein anderes zu verschieben und Objektnamen zu bearbeiten. Weitere Informationen finden Sie unter Erstellen von Migrationsregeln in AWS SCT .	4. Oktober 2016

Version	Änderung	Beschreibung	Datum geändert
1.0.500	Wechsel zur Cloud	Sie können das jetzt verwenden, AWS Schema Conversion Tool um Ihr vorhanden es lokales Datenbankschema auf eine Amazon RDS-DB-Instance zu kopieren, auf der dieselbe Engine ausgeführt wird. Mit dieser Funktion können Sie potenzielle Kosteneinsparungen kalkulieren, die beim Wechsel in die Cloud und einer Änderung des Lizenztyps möglich sind. Weitere Informationen finden Sie unter Erstellung von Berichten zur Migrationsbewertung mitAWS SCT .	4. Oktober 2016
1.0.400	Konvertieren von Data Warehouse-Schemata	Sie können jetzt die verwendenAWS Schema Conversion Tool, um Ihre Data Warehouse-Schemas von Oracle und Teradata nach Amazon Redshift zu konvertieren. Weitere Informationen finden Sie unter Konvertierung von Data Warehouse-Schemas in Amazon Redshift mithilfe von AWS SCT .	13. Juli 2016
1.0.400	Konvertierung des Anwendungs-SQL-Codes	Sie können jetzt mit dem AWS Schema Conversion Tool SQL-Code in Ihren C++-, C #-, Java- oder anderen Anwendungscode konvertieren. Weitere Informationen finden Sie unter Anwendungs-SQL konvertieren mitAWS SCT .	13. Juli 2016

Version	Änderung	Beschreibung	Datum geändert
1.0.400	Neue Funktion	Das enthält AWS Schema Conversion Tool jetzt ein Erweiterungspaket und einen Assistenten, der Sie bei der Installation, Erstellung und Konfiguration von AWS Lambda Funktionen und Python-Bibliotheken unterstützt, um E-Mail, Jobplanung und andere Funktionen bereitzustellen. Weitere Informationen erhalten Sie unter Verwenden Sie die AWS Lambda Funktionen aus dem AWS SCT Erweiterungspaket und Verwenden von benutzerdefinierten Bibliotheken für AWS SCT Erweiterungspakete .	13. Juli 2016
1.0.301	SSL Support	Sie können jetzt mithilfe von Secure Sockets Layer (SSL) eine Verbindung mit Ihrer Quelldatenbank herstellen, wenn Sie das AWS Schema Conversion Tool verwenden.	19. Mai 2016
1.0.203	Neue Funktion	Es wurde eine Unterstützung für MySQL und PostgreSQL als Quelldatenbanken für Konvertierungen hinzugefügt.	11. April 2016
1.0.202	Wartungsversion	Unterstützt die Bearbeitung des konvertierten SQL-Codes, der für die Zieldatenbank-Engine generiert wurde. Implementiert verbesserte Auswahlmöglichkeiten in den Strukturansichten der Quelldatenbank und der Ziel-DB-Instance. Fügt eine Unterstützung für die Verbindung mit einer Oracle-Quelldatenbank über TNS (Transparent Network Substrate)-Namen hinzu.	2. März 2016

Version	Änderung	Beschreibung	Datum geändert
1.0.200	Wartungsversion	Implementiert die Unterstützung von PostgreSQL als Zieldatenbank-Engine hinzu. Ermöglicht das Generieren konvertierter Schemata als Skripts und das Speichern dieser Skripts in Dateien, ehe das Schema auf die Ziel-DB-Instance angewendet wird.	14. Januar 2016
1.0.103	Wartungsversion	Implementiert eine Offline-Projektfunktionalität, die Möglichkeit, nach neuen Versionen zu suchen, und eine Speicher- und Leistungsverwaltung.	2. Dezember 2015
1.0.101	Wartungsversion	Fügt den Assistenten Create New Database Migration Project hinzu. Fügt die Möglichkeit zum Speichern des Bewertungsberichts zur Datenbankmigration als PDF-Datei hinzu.	19. Oktober 2015
1.0.100	Vorversion	Stellt das Benutzerhandbuch für die AWS Schema Conversion Tool-Vorversion zur Verfügung.	7. Oktober 2015

Die vorliegende Übersetzung wurde maschinell erstellt. Im Falle eines Konflikts oder eines Widerspruchs zwischen dieser übersetzten Fassung und der englischen Fassung (einschließlich infolge von Verzögerungen bei der Übersetzung) ist die englische Fassung maßgeblich.