



Panduan Pengguna

AWS Schema Conversion Tool



Versi 1.0.672

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

AWS Schema Conversion Tool: Panduan Pengguna

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

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

Apa AWS SCT?	1
Ringkasan konversi skema	6
Memberikan umpan balik	7
Menginstal, memverifikasi, dan memperbarui	8
Instalasi AWS SCT	8
Menginstal versi sebelumnya	9
Memverifikasi unduhan AWS SCT file	10
Memverifikasi checksum file AWS SCT	10
Memverifikasi file AWS SCT RPM di Fedora	11
Memverifikasi file AWS SCT DEB di Ubuntu	12
Memverifikasi file AWS SCT MSI di Microsoft Windows	12
Mengunduh driver database yang diperlukan	13
Menginstal driver JDBC di Linux	16
Menyimpan jalur driver di pengaturan global	17
Memperbarui AWS SCT	18
AWS SCT CLI	19
Menggunakan antarmuka AWS SCT pengguna	20
Jendela proyek	20
Memulai AWS SCT	22
Membuat proyek	22
Menggunakan wizard proyek baru	23
Menyimpan dan membuka proyek	26
Menambahkan server	27
Menggunakan mode offline	28
Menggunakan filter pohon	29
.....	30
Mengimpor daftar file untuk filter pohon	32
Menyembunyikan skema	33
Mengelola laporan penilaian migrasi database	34
Mengonversi skema Anda	38
Menerapkan kode yang dikonversi	41
Menyimpan AWS profil	42
Menyimpan AWS kredensi	42
Mengatur profil default untuk sebuah proyek	44

Izin untuk menggunakan profilAWS layanan	45
Menggunakan AWS Secrets Manager	46
Menyimpan kata sandi basis data	47
Menggunakan tampilan Union All untuk proyek-proyek dengan tabel yang dipartisi	47
Pintasan keyboard	48
Mulai	50
Sumber untuk AWS SCT	52
Mengkripsi koneksi Amazon RDS	53
Menggunakan Apache Cassandra sebagai sumber	56
Menghubungkan ke Apache Cassandra sebagai sumber	56
Menggunakan Apache Hadoop sebagai sumber	58
Prasyarat untuk menggunakan Apache Hadoop sebagai sumber	59
Izin untuk Hive sebagai sumber	60
Izin untuk HDFS sebagai sumber	60
Izin untuk HDFS sebagai target	61
Menghubungkan ke Apache Hadoop sebagai sumber	61
Menghubungkan ke Hive dan HDFS	63
Menghubungkan ke Amazon EMR sebagai target	66
Menggunakan Apache Oozie sebagai sumber	69
Prasyarat	69
Menghubungkan ke Apache Oozie sebagai sumber	70
Izin untuk AWS Lambda	71
Menghubungkan ke AWS Step Functions sebagai target	73
Menggunakan Azure SQL Database sebagai sumber	75
Hak istimewa untuk Azure SQL Database	75
Menghubungkan ke Azure SQL Database sebagai sumber	75
Menggunakan IBM Db2 untuk z/OS sebagai sumber	77
Prasyarat untuk Db2 untuk z/OS	77
Hak istimewa untuk Db2 untuk z/OS	78
Menghubungkan ke Db2 untuk z/OS sebagai sumber	79
Hak istimewa untuk MySQL sebagai target	81
Hak istimewa untuk PostgreSQL sebagai target	83
Db2 untuk z/OS ke pengaturan konversi PostgreSQL	84
Menggunakan IBM Db2 LUW sebagai sumber	85
Hak Istimewa untuk Db2 LUW	86
Menghubungkan ke Db2 LUW sebagai sumber	88

Db2 LUW para PostgreSQL	90
Db2 LUW para MySQL	93
Menggunakan MySQL sebagai sumber	94
Hak istimewa untuk MySQL	95
Menghubungkan ke MySQL sebagai sumber	95
Hak untuk PostgreSQL sebagai target	98
Menggunakan Oracle Database sebagai sumber	98
Hak istimewa untuk Oracle	99
Menghubungkan ke Oracle sebagai sumber	99
Oracle para PostgreSQL	103
Oracle para MySQL	110
Oracle ke Amazon RDS untuk Oracle	120
Menggunakan PostgreSQL sebagai sumber	127
Hak istimewa untuk PostgreSQL	127
Menghubungkan ke PostgreSQL sebagai sumber	128
Hak istimewa untuk MySQL sebagai target	130
Menggunakan SAP ASE (Sybase ASE) sebagai sumber	131
Keistimewaan untuk SAP ASE	132
Connect ke SAP ASE sebagai sumber	133
Hak istimewa untuk MySQL sebagai target	135
Pengaturan konversi SAP ASE ke MySQL	137
Hak istimewa untuk PostgreSQL sebagai target	137
Pengaturan konversi SAP ASE ke PostgreSQL	138
Menggunakan SQL Server sebagai sumber	139
Hak istimewa untuk Microsoft SQL Server	140
Menggunakan Otentikasi Windows dengan Microsoft SQL Server	141
Menghubungkan ke SQL Server sebagai sumber	143
SQL Server ke MySQL	146
SQL Server ke PostgreSQL	150
Server SQL untuk Amazon RDS SQL Server	186
Sumber gudang data untuk AWS SCT	188
Menggunakan Amazon Redshift sebagai sumber	188
Menggunakan Azure Synapse Analytics sebagai sumber	194
Menggunakan BigQuery sebagai sumber	200
Menggunakan Greenplum Database sebagai sumber	205
Menggunakan Netezza sebagai sumber	211

Menggunakan Oracle Data Warehouse sebagai sumber	221
Menggunakan Snowflake sebagai sumber	228
Menggunakan SQL Server Data Warehouse sebagai sumber	237
Menggunakan Teradata sebagai sumber	243
Menggunakan Vertica sebagai sumber	259
Membuat aturan pemetaan	266
Aturan baru	267
Mengelola aturan	267
Target virtual	268
Keterbatasan:	269
Membuat laporan konversi	271
Laporan penilaian migrasi	271
Membuat laporan penilaian penilaian penilaian penilaian migrasi basis data penilaian migrasi basis data	272
Melihat laporan penilaian	273
Menyimpan laporan penilaian penilaian penilaian penilaian penilaian penilaian penilaian penilaian	277
Mengonfigurasi laporan penilaian penilaian penilaian penilaian penilaian penilaian penilaian penilaian penilaian penilaian	280
Membuat laporan penilaian penilaian penilaian penilaian penilaian penilaian multiserver	283
Mengonversi skema database	292
Membuat aturan migrasi	294
Membuat aturan migrasi	295
Mengekspor aturan migrasi	297
Mengonversi skema	297
Mengubah skema	297
Mengedit skema yang dikonversi	300
Menghapus skema yang dikonversi	301
Menangani konversi manual	301
Memodifikasi skema sumber	301
Memodifikasi skema target	301
Memperbarui dan menyegarkan skema yang telah dikonversi	302
Menyimpan dan menerapkan skema Anda	303
Menyimpan skema yang telah dikonversi	303
Menerapkan skema yang telah dikonversi	304
Skema paket ekstensi	304

Membandingkan skema	305
Objek yang ditransformasikan terkait	307
Mengonversi skema gudang data ke Amazon Redshift	308
Izin untuk Amazon Redshift	309
Memilih strategi dan aturan optimasi	311
Mengumpulkan atau mengunggah statistik	312
Membuat aturan migrasi	314
Membuat aturan migrasi	315
Mengekspor aturan migrasi	316
Mengonversi skema	317
Mengubah skema	317
Mengedit skema yang dikonversi	320
Menghapus skema yang dikonversi	321
Mengelola dan menyesuaikan kunci	321
Topik terkait	322
Membuat dan menggunakan laporan penilaian	322
Membuat laporan penilaian migrasi database	322
Ringkasan	323
Item tindakan	325
Menyimpan laporan penilaian	325
Menangani konversi manual	326
Memodifikasi skema sumber	327
Memodifikasi skema target	327
Memperbarui dan menyegarkan skema yang telah dikonversi	328
Menyimpan dan menerapkan skema yang telah dikonversi	328
Menyimpan skema yang telah dikonversi ke file	329
Menerapkan skema yang telah dikonversi	329
Skema paket ekstensi	330
Pustaka Python	331
Mengoptimalkan Amazon Redshift	331
Mengoptimalkan database Amazon Redshift Anda	331
Mengonversi proses ETL	333
Mengkonversi proses ETL keAWS Glue	334
Prasyarat	335
AWS GlueKatalog Data	336
Keterbatasan:	336

Langkah 1: Buat proyek baru	338
Langkah 2: BuatAWS Gluepekerjaan	339
Mengkonversi proses ETL menggunakan Python API untukAWS Glue	340
Langkah 1: Buat database	341
Langkah 2: Buat koneksi	341
Langkah 3: BuatAWS Glueperayap	343
Mengkonversi skrip ETL Informatica	345
Mengonversi SSIS keAWS Glue	350
Komponen SSIS yang didukung	354
Mengonversi SSIS keAWS Glue Studio	356
Prasyarat	356
Menambahkan paket SSIS keAWS SCTproyek	358
Mengonversi paket SSIS	359
MenciptakanAWS Glue Studiopekerjaan	360
Membuat laporan penilaian konversi SSIS	362
Komponen SSIS yang didukung	362
Mengonversi Teradata BTEQ ke Amazon Redshift RSQL	363
Menambahkan skrip BTEQ keAWS SCTproyek	365
Mengkonfigurasi variabel substitusi dalam skrip BTEQ	365
Mengonversi skrip BTEQ	366
Mengelola skrip BTEQ	367
Membuat laporan penilaian konversi skrip BTEQ	368
Mengedit dan menyimpan skrip BTEQ Anda yang telah dikonversi	368
Mengonversi skrip shell ke Amazon Redshift RSQL	369
Menambahkan skrip shell keAWS SCTproyek	370
Mengkonfigurasi variabel substitusi dalam skrip shell	371
Mengonversi skrip shell	371
Mengelola skrip shell	372
Membuat laporan penilaian konversi skrip shell	373
Mengedit dan menyimpan skrip shell yang telah dikonversi	374
Konversi TeradataFastExportpara Amazon Redshift RSQL	374
MenambahkanFastExportskrip pekerjaan untukAWS SCTproyek	375
Mengkonfigurasi variabel substitusi diFastExportskrip pekerjaan	376
MengonversiFastExportskrip pekerjaan	377
MengelolaFastExportskrip pekerjaan	378
MembuatFastExportlaporan penilaian konversi skrip pekerjaan	378

Mengedit dan menyimpan konversi AndaFastExportskrip pekerjaan	379
Konversi TeradataFastLoadpara Amazon Redshift RSQL	379
MenambahkanFastLoadskrip pekerjaan untuk AndaAWS SCTproyek	380
Mengkonfigurasi variabel substitusi diFastLoadskrip pekerjaan	381
MengonversiFastLoadskrip pekerjaan	383
MengelolaFastLoadskrip pekerjaan	384
MembuatFastLoadlaporan penilaian konversi skrip pekerjaan	384
Mengedit dan menyimpan konversi AndaFastLoadskrip pekerjaan	385
Konversi TeradataMultiLoadpara Amazon Redshift RSQL	386
MenambahkanMultiLoadskrip pekerjaan untuk AndaAWS SCTproyek	386
Mengkonfigurasi variabel substitusi diMultiLoadskrip pekerjaan	387
MengonversiMultiLoadskrip pekerjaan	388
MengelolaMultiLoadskrip pekerjaan	389
MenciptakanMultiLoadlaporan penilaian konversi skrip pekerjaan	390
Mengedit dan menyimpan konversi AndaMultiLoadskrip pekerjaan	391
Memigrasi kerangka kerja data besar	392
Memigrasi Apache Hadoop ke Amazon EMR	392
Gambaran Umum	393
Langkah 1: Hubungkan ke kluster Hadoop Anda	394
Langkah 2: Mengatur aturan pemetaan	394
Langkah 3: Buat laporan penilaian	396
Langkah 4: Migrasikan kluster Apache Hadoop Anda ke Amazon EMR	397
Menjalankan skrip CLI Anda	398
Mengelola proyek migrasi	399
Mengkonversi Apache Oozie ke AWS Step Functions	401
Gambaran Umum	401
Langkah 1: Hubungkan ke sumber dan layanan target Anda	403
Langkah 2: Mengatur aturan pemetaan	403
Langkah 3: Konfigurasi parameter	404
Langkah 4: Buat laporan penilaian	406
Langkah 5: Konversikan alur kerja Apache Oozie Anda ke AWS Step Functions	407
Menjalankan skrip CLI Anda	410
Node yang didukung	410
Menggunakan AWS SCT dengan AWS DMS	412
MenggunakanAWS SCTagen replikasi denganAWS DMS	412
MenggunakanAWS SCTagen ekstraksi data denganAWS DMS	412

Meningkatkan tingkat logging saat menggunakan AWS SCT bersama AWS DMS	412
Migrasi dari gudang data ke Amazon Redshift	414
Prasyarat	416
Pengaturan Amazon S3	417
Dengan asumsi peran IAM	418
Pengaturan keamanan	419
Pengaturan konfigurasi	420
Instalasi agen	420
Mengkonfigurasi agen	422
Menginstal dan mengonfigurasi agen penyalinan khusus	424
Agen awal	426
Mendaftarkan agen	427
Menyembunyikan dan memulihkan informasi untuk agen AWS SCT	427
Membuat aturan migrasi data	429
Mengubah setelan ekstraktor dan salin untuk migrasi data	430
Menyortir data	433
Membuat, menjalankan, dan memantau AWS SCT tugas	435
Mengekspor dan mengimpor tugas ekstraksi data	438
Ekstraksi data menggunakan perangkat AWS Snowball Edge	439
tep-by-step Prosedur S untuk memigrasi data menggunakan AWS SCT dan AWS Snowball Edge	440
Output tugas ekstraksi data	443
Menggunakan partisi virtual	445
Batas saat membuat partisi virtual	445
Jenis partisi RANGE	446
LIST jenis partisi	447
Jenis partisi DATE AUTO SPLIT	448
Menggunakan partisi asli	449
Bekerja dengan LOB	450
Praktik terbaik dan pemecahan masalah	451
Mengonversi aplikasi SQL	453
Ikhtisar mengkonversi aplikasi SQL	453
Mengonversi kode SQL dalam aplikasi Anda	454
Membuat proyek konversi aplikasi generik	454
Mengelola proyek konversi aplikasi	458
Menganalisis dan mengonversi kode SQL	459

Membuat dan menggunakan laporan penilaian	460
Mengedit dan menyimpan kode SQL Anda yang telah dikonversi	461
Mengonversi kode SQL dalam aplikasi C #	461
Membuat proyek konversi aplikasi C #	462
Mengonversi kode SQL aplikasi C #	463
Menyimpan kode aplikasi Anda yang telah dikonversi	465
Mengelola proyek konversi aplikasi C #	465
Membuat laporan penilaian konversi aplikasi C#	466
Mengonversi kode SQL dalam aplikasi C ++	467
Membuat proyek konversi aplikasi C ++	467
Mengonversi kode SQL aplikasi C++ Anda	469
Menyimpan kode aplikasi Anda yang telah dikonversi	470
Mengelola proyek konversi aplikasi C ++	471
Membuat laporan penilaian konversi aplikasi C++	472
Mengonversi kode SQL dalam aplikasi Java	473
Membuat proyek konversi aplikasi Java	474
Mengonversi kode SQL aplikasi Java Anda	476
Menyimpan kode aplikasi Anda yang telah dikonversi	478
Mengelola proyek konversi aplikasi Java	478
Membuat laporan penilaian konversi aplikasi Java	479
Mengonversi kode SQL dalam aplikasi Pro* C	480
Membuat proyek konversi aplikasi Pro* C	481
Mengonversi kode SQL aplikasi Pro* C Anda	482
Mengedit dan menyimpan kode aplikasi Anda yang telah dikonversi	484
Mengelola proyek konversi aplikasi Pro* C	485
Membuat laporan penilaian konversi aplikasi Pro*C	485
Menggunakan paket ekstensi	488
Izin untuk menggunakan paket ekstensi	489
Menggunakan skema paket ekstensi	490
Pustaka khusus untuk paket ekstensi	491
Menerapkan paket ekstensi	491
Menggunakan fungsi Lambda dari paket ekstensi AWS SCT	494
Menggunakan AWS Lambda fungsi untuk meniru fungsionalitas database	494
Menerapkan paket ekstensi untuk mendukung fungsi Lambda	494
Mengkonfigurasi fungsi paket ekstensi	496
Praktik terbaik	498

Mengkonfigurasi memori tambahan	498
folder proyek default	498
Meningkatkan kecepatan migrasi data	499
Meningkatkan informasi logging	499
Pemecahan Masalah	502
Tidak dapat memuat objek dari sumber basis data Oracle	502
Pesan WARWARR WARR WAR	502
Referensi CLI	504
Prasyarat	504
Mode interaktif	504
Contoh-contoh	506
Mendapatkan skenario CLI	506
Contoh-contoh	510
Mengedit skenario CLI	510
Mode skrip	512
Contoh-contoh	512
Bahan referensi	513
Catatan rilis	514
Catatan rilis — 675	514
Catatan rilis — 674	516
Catatan rilis — 673	523
Catatan rilis — 672	528
Catatan rilis — 671	537
Catatan rilis — 670	546
Catatan rilis — 669	551
Catatan rilis — 668	556
Catatan rilis — 667	562
Catatan rilis — 666	566
Catatan rilis — 665	571
Catatan rilis — 664	574
Catatan rilis — 663	578
Catatan rilis — 662	581
Catatan rilis — 661	586
Catatan rilis — 660	590
Catatan rilis — 659	594
Catatan rilis — 658	598

Catatan rilis — 657	604
Catatan rilis — 656	608
Catatan rilis — 655	611
Catatan rilis — 654	614
Catatan rilis — 653	617
Catatan rilis — 652	619
Catatan rilis — 651	622
Catatan rilis — 650	624
Catatan rilis — 649	626
Catatan rilis — 648	629
Catatan rilis — 647	630
Catatan rilis — 646	632
Catatan rilis — 645	634
Catatan rilis — 644	635
Catatan rilis — 642	637
Catatan rilis — 641	639
Catatan rilis — 640	640
Rilis 1.0.640 perubahan Oracle	640
Rilis 1.0.640 Microsoft SQL Server perubahan	646
Rilis 1.0.640 Perubahan MySQL	650
Rilis 1.0.640 perubahan PostgreSQL	651
Rilis 1.0.640 Db2 perubahan LUW	654
Rilis 1.0.640 Perubahan Teradata	655
Rilis 1.0.640 perubahan untuk mesin lain	657
Riwayat dokumen	660
Pembaruan sebelumnya	676
.....	dclxxxiv

Apa itu AWS Schema Conversion Tool?

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengkonversi skema database yang ada dari satu mesin database yang lain. Anda dapat mengkonversi skema OLTP relasional, atau skema gudang data. Skema yang dikonversi cocok untuk Amazon Relational Database Service (Amazon RDS) MySQL, MariaDB, Oracle, SQL Server, PostgreSQL DB, klaster Amazon Aurora DB, atau klaster Amazon Redshift. Skema yang dikonversi juga dapat digunakan dengan database pada instans Amazon EC2 atau disimpan sebagai data pada bucket Amazon S3.

AWS SCT mendukung beberapa standar industri, termasuk Federal Information Processing Standards (FIPS), untuk koneksi ke bucket Amazon S3 atau sumber daya lain AWS. AWS SCT juga sesuai dengan Program Manajemen Risiko dan Otorisasi Federal (FedRAMP). Untuk detail tentang AWS dan upaya kepatuhan, lihat [layanan AWS dalam lingkup berdasarkan program kepatuhan](#).

AWS SCT mendukung konversi OLTP berikut.

Basis data sumber	Target database
IBM Db2 untuk z/OS (versi 12)	Edisi Kompatibel dengan Amazon Aurora MySQL (Aurora MySQL), Edisi Kompatibel dengan Amazon Aurora PostgreSQL (Aurora PostgreSQL), MySQL, PostgreSQL Untuk informasi selengkapnya, lihat Menggunakan IBM Db2 untuk z/OS sebagai sumber .
IBM Db2 LUW (versi 9.1, 9.5, 9.7, 10.5, 11.1, dan 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL Untuk informasi selengkapnya, lihat Menggunakan IBM Db2 LUW sebagai sumber .
Database Microsoft Azure SQL	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL

Basis data sumber	Target database
	Untuk informasi selengkapnya, lihat Menggunakan Azure SQL Database sebagai sumber .
Microsoft SQL Server (versi 2008 R2, 2012, 2014, 2016, 2017, 2019, dan 2022)	Aurora MySQL, Aurora PostgreSQL, Babelfish untuk Aurora PostgreSQL (hanya untuk laporan penilaian), MariaDB, Microsoft SQL Server, MySQL, PostgreSQL Untuk informasi selengkapnya, lihat Menggunakan SQL Server sebagai sumber .
MySQL (versi 5.5 dan lebih tinggi)	Aurora PostgreSQL, MySQL, PostgreSQL Untuk informasi selengkapnya, lihat Menggunakan MySQL sebagai sumber . Anda dapat memigrasikan skema dan data dari MySQL ke cluster Aurora MySQL DB tanpa menggunakan AWS SCT Untuk informasi selengkapnya, lihat Memigrasi data ke kluster DB Amazon Aurora .
Oracle (versi 10.1 dan lebih tinggi)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL Untuk informasi selengkapnya, lihat Menggunakan Oracle Database sebagai sumber .
PostgreSQL (versi 9.1 dan lebih tinggi)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL Untuk informasi selengkapnya, lihat Menggunakan PostgreSQL sebagai sumber .

Basis data sumber	Target database
SAP ASE (versi 12.5.4, 15.0.2, 15.5, 15.7, dan 16.0)	<p>Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL</p> <p>Untuk informasi selengkapnya, lihat Menggunakan SAP ASE (Sybase ASE) sebagai sumber.</p>

AWS SCT mendukung konversi data warehouse berikut.

Sumber data gudang	Target gudang data
Amazon Redshift	<p>Amazon Redshift</p> <p>Untuk informasi selengkapnya, lihat Menggunakan Amazon Redshift sebagai sumber.</p>
Analisis Azure Synapse	<p>Amazon Redshift</p> <p>Untuk informasi selengkapnya, lihat Menggunakan Azure Synapse Analytics sebagai sumber.</p>
BigQuery	<p>Amazon Redshift</p> <p>Untuk informasi selengkapnya, lihat Menggunakan BigQuery sebagai sumber.</p>
Database Greenplum (versi 4.3 dan 6.21)	<p>Amazon Redshift</p> <p>Untuk informasi selengkapnya, lihat Menggunakan Greenplum Database sebagai sumber.</p>
Microsoft SQL Server (versi 2008 dan yang lebih tinggi)	<p>Amazon Redshift</p>

Sumber data gudang	Target gudang data
	Untuk informasi selengkapnya, lihat Menggunakan SQL Server Data Warehouse sebagai sumber .
Netezza (versi 7.0.3 dan lebih tinggi)	Amazon Redshift Untuk informasi selengkapnya, lihat Menggunakan Netezza sebagai sumber .
Oracle (versi 10.1 dan lebih tinggi)	Amazon Redshift Untuk informasi selengkapnya, lihat Menggunakan Oracle Data Warehouse sebagai sumber .
Kepingan salju (versi 3)	Amazon Redshift Untuk informasi selengkapnya, lihat Menggunakan Snowflake sebagai sumber .
Teradata (versi 13 dan lebih tinggi)	Amazon Redshift Untuk informasi selengkapnya, lihat Menggunakan Teradata sebagai sumber .
Vertica (versi 7.2.2 dan lebih tinggi)	Amazon Redshift Untuk informasi selengkapnya, lihat Menggunakan Vertica sebagai sumber .

AWS SCT mendukung konversi database data NoSQL berikut.

Basis data sumber	Target database
Apache Cassandra (versi 2.1.x, 2.2.16, dan 3.11.x)	Amazon DynamoDB

Basis data sumber	Target database
	Untuk informasi selengkapnya, lihat Menggunakan Apache Cassandra sebagai sumber .

AWS SCT mendukung konversi ekstrak berikut, transformasi, dan beban (ETL) proses. Untuk informasi selengkapnya, lihat [Mengonversi proses ETL](#).

Sumber	Target
Skrip ETL Informatica	Informatika
Paket ETL Microsoft SQL Server Integrasi Services (SSIS)	AWS Glue atau AWS Glue Studio
Script shell dengan perintah tertanam dari Teradata Dasar Teradata Query (BTEQ)	Amazon Redshift
Skrip ETL Teradata BTEQ	AWS Glue atau Amazon Redshift RSQL
Script pekerjaan Teradata FastExport	Amazon Redshift
Script pekerjaan Teradata FastLoad	Amazon Redshift
Script pekerjaan Teradata MultiLoad	Amazon Redshift

AWS SCT mendukung migrasi kerangka data besar berikut. Untuk informasi selengkapnya, lihat [Memigrasi kerangka kerja data besar](#).

Sumber	Target
Apache Hive (versi 0.13.0 dan lebih tinggi)	Sarang di Amazon EMR
HDFS Apache	Amazon S3 atau HDFS di Amazon EMR
Apache Oozie	AWS Step Functions

Ringkasan konversi skema

AWS SCT menyediakan antarmuka pengguna berbasis proyek untuk secara otomatis mengonversi skema database sumber Anda ke dalam format yang kompatibel dengan instans Amazon RDS target Anda. Jika skema dari database sumber Anda tidak dapat dikonversi secara otomatis, AWS SCT berikan panduan tentang bagaimana Anda dapat membuat skema yang setara dalam database Amazon RDS target Anda.

Untuk informasi tentang cara menginstal AWS SCT, lihat [Menginstal, memverifikasi, dan memperbarui AWS SCT](#).

Untuk pengenalan antarmuka AWS SCT pengguna, lihat [Menggunakan antarmuka AWS SCT pengguna](#).

Untuk informasi tentang proses konversi, lihat [Mengonversi skema database menggunakan AWS SCT](#).

Selain mengkonversi skema database yang ada dari satu mesin database yang lain, AWS SCT memiliki beberapa fitur tambahan yang membantu Anda memindahkan data dan aplikasi Anda ke Cloud: AWS

- Anda dapat menggunakan agen ekstraksi data untuk mengekstrak data dari gudang data Anda guna mempersiapkan migrasi ke Amazon Redshift. Untuk mengelola agen ekstraksi data, Anda dapat menggunakan AWS SCT. Untuk informasi selengkapnya, lihat [Memigrasi data dari gudang data lokal ke Amazon Redshift](#).
- Anda dapat menggunakan AWS SCT untuk membuat AWS DMS titik akhir dan tugas. Anda dapat menjalankan dan memantau tugas-tugas ini dari AWS SCT. Untuk informasi selengkapnya, lihat [Menggunakan AWS SCT dengan AWS DMS](#).
- Dalam beberapa kasus, fitur database tidak dapat dikonversi ke fitur Amazon RDS atau Amazon Redshift yang setara. Wisaya paket AWS SCT ekstensi dapat membantu Anda menginstal AWS Lambda fungsi dan pustaka Python untuk meniru fitur yang tidak dapat dikonversi. Untuk informasi selengkapnya, lihat [Menggunakan paket AWS SCT ekstensi](#).
- Anda dapat menggunakan AWS SCT untuk mengoptimalkan database Amazon Redshift yang ada. AWS SCT merekomendasikan kunci sortir dan kunci distribusi untuk mengoptimalkan database Anda. Untuk informasi selengkapnya, lihat [Mengoptimalkan Amazon Redshift dengan menggunakan AWS SCT](#).

- Anda dapat menggunakannya AWS SCT untuk menyalin skema database lokal yang ada ke instans DB Amazon RDS yang menjalankan mesin yang sama. Anda dapat menggunakan fitur ini untuk menganalisis potensi penghematan biaya pindah ke cloud dan mengubah jenis lisensi Anda.
- Anda dapat menggunakan AWS SCT untuk mengkonversi SQL di C ++, C #, Java, atau kode aplikasi lainnya. Anda dapat melihat, menganalisis, mengedit, dan menyimpan kode SQL yang dikonversi. Untuk informasi selengkapnya, lihat [Mengonversi aplikasi SQL menggunakan AWS SCT](#).
- Anda dapat menggunakan AWS SCT untuk memigrasi proses ekstraksi, transformasi, dan beban (ETL). Untuk informasi selengkapnya, lihat [Mengkonversi ekstrak, transformasi, dan beban \(ETL\) proses dengan AWS Schema Conversion Tool](#).

Memberikan umpan balik

Anda dapat memberikan umpan balik tentang AWS SCT. Anda dapat mengajukan laporan bug, mengirimkan permintaan fitur, atau memberikan informasi umum.

Untuk memberikan umpan balik tentang AWS SCT

1. Memulai AWS Schema Conversion Tool.
2. Buka menu Bantuan dan kemudian pilih Tinggalkan Umpan Balik. Kotak dialog Tinggalkan Umpan Balik muncul.
3. Untuk Area, pilih Informasi, Laporan bug, atau Permintaan fitur.
4. Untuk database Sumber, pilih database sumber Anda. Pilih Apa saja jika umpan balik Anda tidak spesifik untuk database tertentu.
5. Untuk database Target, pilih database target Anda. Pilih Apa saja jika umpan balik Anda tidak spesifik untuk database tertentu.
6. Untuk Judul, ketik judul untuk umpan balik Anda.
7. Untuk Pesan, ketik umpan balik Anda.
8. Pilih Kirim untuk mengirimkan umpan balik Anda.

Menginstal, memverifikasi, dan memperbarui AWS SCT

The AWS Schema Conversion Tool (AWS SCT) adalah aplikasi mandiri yang menyediakan antarmuka pengguna berbasis proyek. AWS SCT tersedia untuk Microsoft Windows, Fedora Linux, dan Ubuntu Linux. AWS SCT hanya didukung pada sistem operasi 64-bit.

Untuk memastikan bahwa Anda mendapatkan versi yang benar dari file AWS SCT distribusi, kami menyediakan langkah-langkah verifikasi setelah Anda mengunduh file terkompresi. Anda dapat memverifikasi file menggunakan langkah-langkah yang disediakan.

Topik

- [Instalasi AWS SCT](#)
- [Memverifikasi unduhan AWS SCT file](#)
- [Mengunduh driver database yang diperlukan](#)
- [Memperbarui AWS SCT](#)
- [AWS SCT CLI](#)

Instalasi AWS SCT

Anda dapat menginstal AWS SCT pada sistem operasi berikut:

- Microsoft Windows 10
- Fedora Linux 36 dan lebih tinggi
- Ubuntu Linux 18 dan lebih tinggi

Untuk menginstal AWS SCT

1. Unduh file terkompresi yang berisi AWS SCT penginstal, menggunakan tautan untuk sistem operasi Anda. Semua file terkompresi memiliki ekstensi.zip. Ketika Anda mengekstrak file AWS SCT installer, itu akan dalam format yang sesuai untuk sistem operasi Anda.

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

- Ekstrak file AWS SCT installer untuk sistem operasi Anda, ditampilkan berikut.

Sistem operasi	Nama file
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

- Jalankan file AWS SCT installer yang diekstrak pada langkah sebelumnya. Gunakan instruksi untuk sistem operasi Anda, yang ditunjukkan berikut.

Sistem operasi	Instal instruksi
Fedora Linux	Jalankan perintah berikut di folder tempat Anda mengunduh file: sudo yum install aws-schema-conversion-tool-1.0. <i>build-number</i> .x86_64.rpm
Microsoft Windows	Klik dua kali file untuk menjalankan penginstal.
Ubuntu Linux	Jalankan perintah berikut di folder tempat Anda mengunduh file: sudo dpkg -i aws-schema-conversion-tool-1.0. <i>build-number</i> .deb

- Unduh driver Java Database Connectivity (JDBC) untuk mesin basis data sumber dan target Anda. Untuk petunjuk dan tautan unduhan, lihat [Mengunduh driver database yang diperlukan](#).

Sekarang, Anda telah menyelesaikan pengaturan AWS SCT aplikasi. Klik dua kali ikon aplikasi untuk menjalankan AWS SCT.

Menginstal versi sebelumnya AWS SCT

Anda dapat mengunduh dan menginstal versi sebelumnya AWS SCT mulai dari 1.0.625. Untuk mengunduh versi sebelumnya, berikan versi dan informasi OS menggunakan format berikut.

```
https://d211wdu1froga6.cloudfront.net/builds/1.0/<version>/<OS>/aws-schema-conversion-tool-1.0.zip
```

Misalnya, untuk mengunduh AWS SCT versi 668, lakukan hal berikut:

- Windows - <https://d211wdu1froga6.cloudfront.net/builds/1.0/668/Windows/ aws-schema-conversion-tool -1.0.zip>
- Ubuntu - <https://d211wdu1froga6.cloudfront.net/builds/1.0/668/Ubuntu/ aws-schema-conversion-tool -1.0.zip>
- Fedora - <https://d211wdu1froga6.cloudfront.net/builds/1.0/668/Fedora/ -1.0.zip aws-schema-conversion-tool>

AWS mendukung tiga versi terbaru dari AWS SCT aplikasi. Kami menyarankan Anda menggunakan versi terbaru AWS SCT.

Memverifikasi unduhan AWS SCT file

Ada beberapa cara Anda dapat memverifikasi file distribusi AWS SCT. Yang paling sederhana adalah membandingkan checksum file dengan checksum yang diterbitkan dari AWS. Sebagai tingkat keamanan tambahan, Anda dapat menggunakan prosedur berikut untuk memverifikasi file distribusi, berdasarkan sistem operasi tempat Anda menginstal file.

Bagian ini mencakup topik berikut.

Topik

- [Memverifikasi checksum file AWS SCT](#)
- [Memverifikasi file AWS SCT RPM di Fedora](#)
- [Memverifikasi file AWS SCT DEB di Ubuntu](#)
- [Memverifikasi file AWS SCT MSI di Microsoft Windows](#)

Memverifikasi checksum file AWS SCT

Untuk mendeteksi kesalahan apa pun yang mungkin terjadi saat mengunduh atau menyimpan file AWS SCT terkompresi, Anda dapat membandingkan checksum file dengan nilai yang disediakan oleh AWS. AWS menggunakan algoritma SHA256 untuk checksum.

Untuk memverifikasi file AWS SCT distribusi menggunakan checksum

1. Unduh file AWS SCT distribusi menggunakan tautan di bagian Instalasi. Untuk informasi selengkapnya, lihat [Instalasi AWS SCT](#).
2. Unduh file checksum terbaru, yang disebut [sha256Check.txt](#). File ini menyertakan checksum untuk versi terbaru AWS SCT . Misalnya, file dapat muncul sebagai berikut:

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

3. Jalankan perintah validasi SHA256 untuk sistem operasi Anda di direktori yang berisi file distribusi. Misalnya, jalankan perintah berikut di Linux.

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

4. Bandingkan hasil perintah dengan nilai yang ditunjukkan dalam file sha256Check.txt. Jika checksum cocok, maka aman untuk menjalankan file distribusi. Jika checksum tidak cocok, maka jangan jalankan file distribusi, dan hubungi [AWS Support](#).

Memverifikasi file AWS SCT RPM di Fedora

AWS memberikan tingkat validasi lain selain checksum file distribusi. Semua file RPM dalam file distribusi ditandatangani oleh kunci AWS pribadi. Kunci GPG publik dapat dilihat di [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key).

Untuk memverifikasi file AWS SCT RPM di Fedora

1. Unduh file AWS SCT distribusi menggunakan tautan di bagian Instalasi.
2. Verifikasi checksum file AWS SCT distribusi.
3. Ekstrak isi file distribusi. Temukan file RPM yang ingin Anda verifikasi.
4. [Unduh kunci publik GPG dari amazon.com.public.gpg-key](#)
5. Impor kunci publik ke RPM DB Anda (pastikan Anda memiliki izin yang sesuai) dengan menggunakan perintah berikut:


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

6. Periksa apakah impor berhasil dengan menjalankan perintah berikut:

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

7. Periksa tanda tangan RPM dengan menjalankan perintah berikut:

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

Memverifikasi file AWS SCT DEB di Ubuntu

AWS memberikan tingkat validasi lain selain checksum file distribusi. Semua file DEB dalam file distribusi ditandatangani oleh tanda tangan terpisah GPG.

Untuk memverifikasi file AWS SCT DEB di Ubuntu

1. Unduh file AWS SCT distribusi menggunakan tautan di bagian Instalasi.
2. Memverifikasi checksum file AWS SCT distribusi.
3. Ekstrak isi file distribusi. Temukan file DEB yang ingin Anda verifikasi.
4. Unduh tanda tangan terpisah dari [aws-schema-conversion-tool-1.0.latest.deb.asc](https://aws-schemas.amazonaws.com/latest/deb/asc).
5. Unduh kunci publik GPG dari [amazon.com.public.gpg-key](https://aws-schemas.amazonaws.com/public/gpg-key).
6. Impor kunci publik GPG dengan menjalankan perintah berikut:

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

7. Verifikasi tanda tangan dengan menjalankan perintah berikut:

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

Memverifikasi file AWS SCT MSI di Microsoft Windows

AWS memberikan tingkat validasi lain selain checksum file distribusi. File MSI memiliki tanda tangan digital yang dapat Anda periksa untuk memastikannya ditandatangani oleh AWS.

Untuk memverifikasi file AWS SCT MSI pada Windows

1. Unduh file AWS SCT distribusi menggunakan tautan di bagian Instalasi.
2. Memverifikasi checksum file AWS SCT distribusi.
3. Ekstrak isi file distribusi. Temukan file MSI yang ingin Anda verifikasi.
4. Di Windows Explorer, klik kanan file MSI dan pilih Properties.
5. Pilih tab Tanda Tangan Digital.
6. Verifikasi bahwa tanda tangan digital berasal dari Amazon Services LLC.

Mengunduh driver database yang diperlukan

AWS SCT Agar berfungsi dengan benar, unduh driver JDBC untuk mesin basis data sumber dan target Anda. Jika Anda menggunakan platform basis data target virtual, Anda tidak perlu mengunduh driver JDBC untuk mesin basis data target Anda. Untuk informasi selengkapnya, lihat [Menggunakan target virtual](#).

Setelah Anda mengunduh driver, Anda memberikan lokasi file driver. Untuk informasi selengkapnya, lihat [Menyimpan jalur driver di pengaturan global](#).

Anda dapat mengunduh driver database dari lokasi berikut.

Important

Unduh versi terbaru dari driver yang tersedia. Tabel berikut mencakup versi terendah dari driver database yang didukung oleh AWS SCT.

Mesin basis data	Pengemudi	Lokasi unduhan
Edisi yang Kompatibel dengan Amazon Aurora MySQL	mysql-connector-java-5.1.6.jar	https://www.mysql.com/products/connector/

Mesin basis data	Pengemudi	Lokasi unduhan
Edisi yang Kompatibel dengan Amazon Aurora PostgreSQL	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
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 Tanpa Server	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
Database Azure SQL	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-driver?dilihat=sql-server-ver-15-#72
Analisis Sinaps Azure	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-driver?dilihat=sql-server-ver-15-#72
Database Greenplum	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
IBM Db2 untuk z/OS	db2jcc-db2jcc4.jar	https://www.ibm.com/support/pages/db2-jdbc-driver-versions-and-download-db2-zos

Mesin basis data	Pengemudi	Lokasi unduhan
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
Microsoft SQL Server	mssql-jdbc-10.2.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/download-microsoft-jdbc-driver-for-sql-server?melihat=15sql-server-ver
MySQL	mysql-connector-java-8.0.15.jar	https://dev.mysql.com/downloads/connector/j/
Netezza	nzjdbc.jar Gunakan perangkat lunak alat klien. Unduh driver versi 7.2.1, yang kompatibel dengan gudang data versi 7.2.0.	http://www.ibm.com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.datacon.doc/c_datacon_plg_overview.html
Oracle	ojdbc8.jar Driver versi 8 dan lebih tinggi didukung.	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	https://dcx.sap.com/qla170/en/html/3bd588c76c5f10149824867aa281fadd.html

Mesin basis data	Pengemudi	Lokasi unduhan
Kepingan salju	<p>snowflake-jdbc-3.9.2.jar</p> <p>Untuk informasi selengkapnya, lihat Mengunduh/ Mengintegrasikan Driver JDBC.</p>	<p>https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9.2.jar</p>
Teradata	<p>terajdbc4.jar</p> <p>tdgssconfig.jar</p> <p>Untuk driver Teradata JDBC versi 16.20.00.11 dan yang lebih tinggi, Anda tidak memerlukan file. tdgssconfig.jar</p>	<p>https://downloads.teradata.com/download/connectivity/jdbc-driver</p>
Vertica	<p>vertica-jdbc-9.1.1-0.jar</p> <p>Versi driver 7.2.0 dan yang lebih tinggi didukung.</p>	<p>https://www.vertica.com/client_drivers/9.1.x/9.1.1-0/vertica-jdbc-9.1.1-0.jar</p>

Menginstal driver JDBC di Linux

Anda dapat menggunakan langkah-langkah berikut untuk menginstal driver JDBC pada sistem Linux Anda untuk digunakan dengan. AWS SCT

Untuk menginstal driver JDBC di sistem Linux Anda

1. Buat direktori untuk menyimpan driver JDBC di.

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

2. Instal driver JDBC untuk mesin database Anda menggunakan perintah yang ditunjukkan berikut.

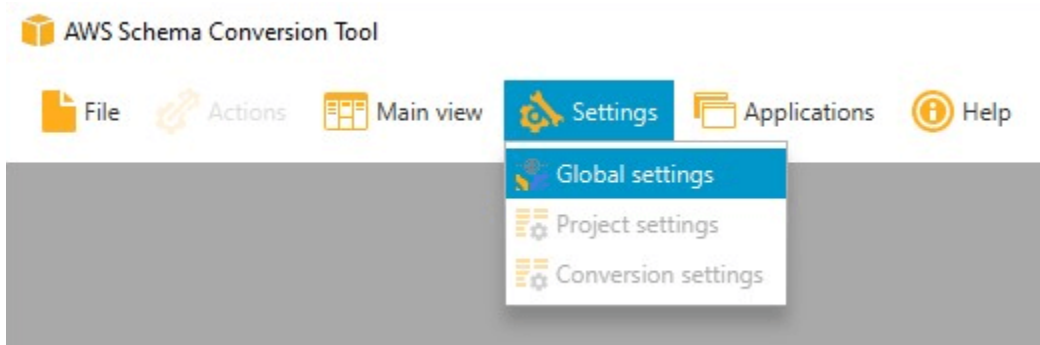
Mesin basis data	Perintah instalasi
Amazon Aurora (kompatibel dengan MySQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>
Amazon Aurora (PostgreSQL kompatibel)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</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>

Menyimpan jalur driver di pengaturan global

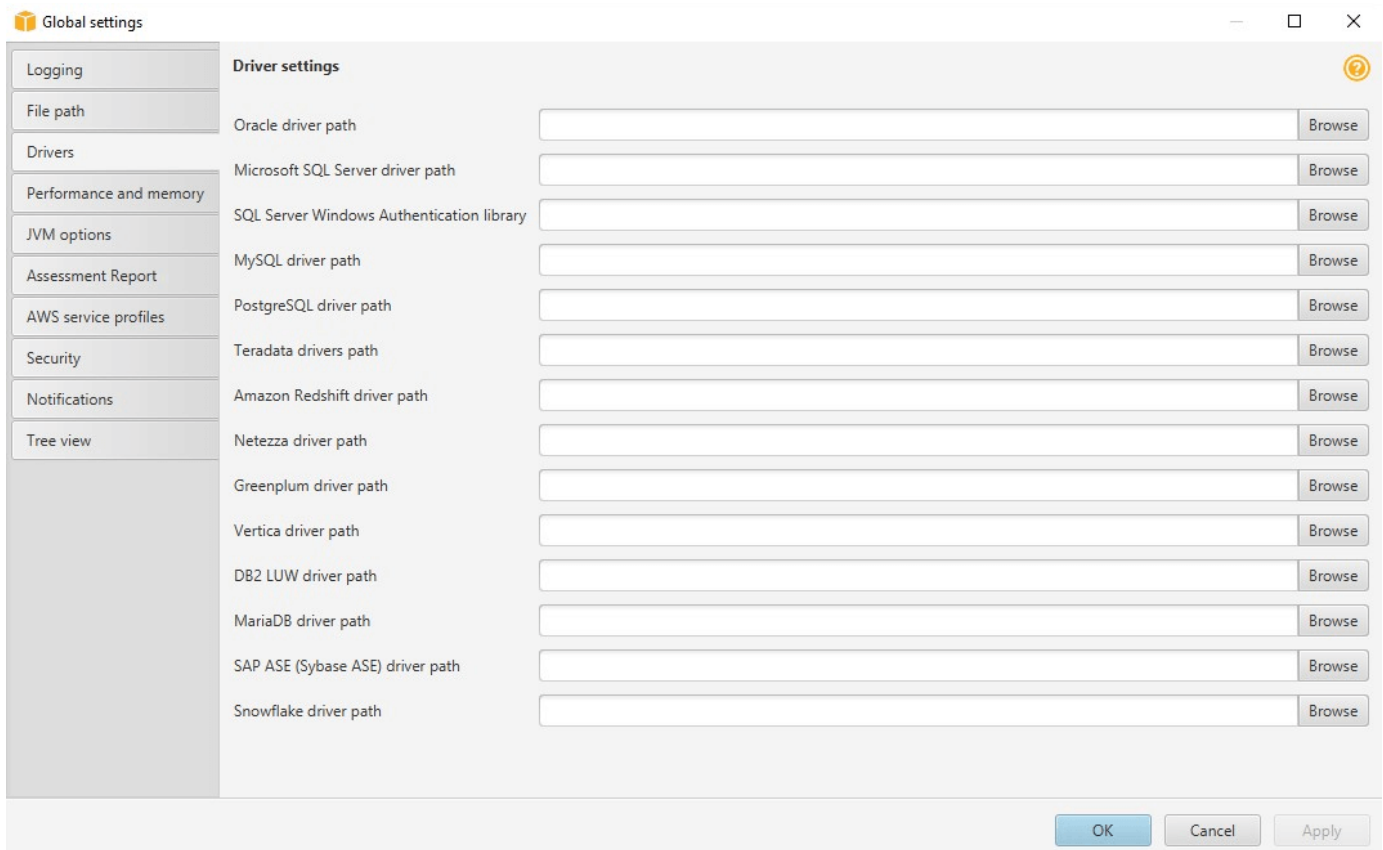
Setelah Anda mengunduh dan menginstal driver JDBC yang diperlukan, Anda dapat mengatur lokasi driver secara global dalam pengaturan. AWS SCT Jika Anda tidak mengatur lokasi driver secara global, aplikasi akan meminta lokasi driver saat Anda terhubung ke database.

Untuk memperbarui lokasi file driver

1. Di AWS SCT, pilih Pengaturan, lalu pilih Pengaturan Global.



- Untuk pengaturan Global, pilih Driver. Tambahkan path file ke driver JDBC untuk mesin database sumber Anda dan mesin database instans Amazon RDS DB target Anda.



- Setelah selesai menambahkan jalur driver, pilih OK.

Memperbarui AWS SCT

AWS memperbarui secara berkala AWS SCT dengan fitur dan fungsionalitas baru. Jika Anda memperbarui dari versi sebelumnya, buat AWS SCT proyek baru dan ubah kembali objek database apa pun yang Anda gunakan.

Anda dapat memeriksa untuk melihat apakah ada pembaruan untuk AWS SCT.

Untuk memeriksa pembaruan AWS SCT

1. Saat masuk AWS SCT, pilih Bantuan dan kemudian pilih Periksa Pembaruan.
2. Dalam kotak dialog Periksa Pembaruan, pilih Apa yang Baru. Jika tautan tidak muncul, Anda memiliki versi terbaru.

AWS SCT CLI

Anda dapat mengunduh AWS SCT CLI untuk penggunaan baris perintah. Untuk mengunduh JAR, gunakan tautan berikut:

[AWSSchemaConversionToolBatch.jar](#)

Menggunakan antarmukaAWS SCT pengguna

Gunakan topik-topik berikut untuk membantu Anda menggunakan antarmukaAWS SCT pengguna. Untuk informasi tentang menginstalAWS SCT, lihat[Menginstal, memverifikasi, dan memperbarui AWS SCT](#).

Topik

- [JendelaAWS SCT proyek](#)
- [MemulaiAWS SCT](#)
- [MembuatAWS SCT proyek](#)
- [Menggunakan wizard proyek baru diAWS SCT](#)
- [Menyimpan dan membukaAWS SCT proyek](#)
- [Menambahkan server database keAWS SCT proyek](#)
- [BerjalanAWS SCT dalam mode offline](#)
- [Menggunakan filterAWS SCT pohon](#)
- [Menyembunyikan skema dalam tampilanAWS SCT pohon](#)
- [Membuat dan meninjau laporan penilaian migrasi database](#)
- [Mengonversi skema Anda](#)
- [Menerapkan skema yang dikonversi ke instans DB target Anda](#)
- [Menyimpan profilAWS layanan diAWS SCT](#)
- [Menggunakan AWS Secrets Manager](#)
- [Menyimpan kata sandi basis data](#)
- [Menggunakan tampilan UNION ALL untuk proyek-proyek dengan tabel dipartisi](#)
- [Pintasan keyboard untukAWS SCT](#)

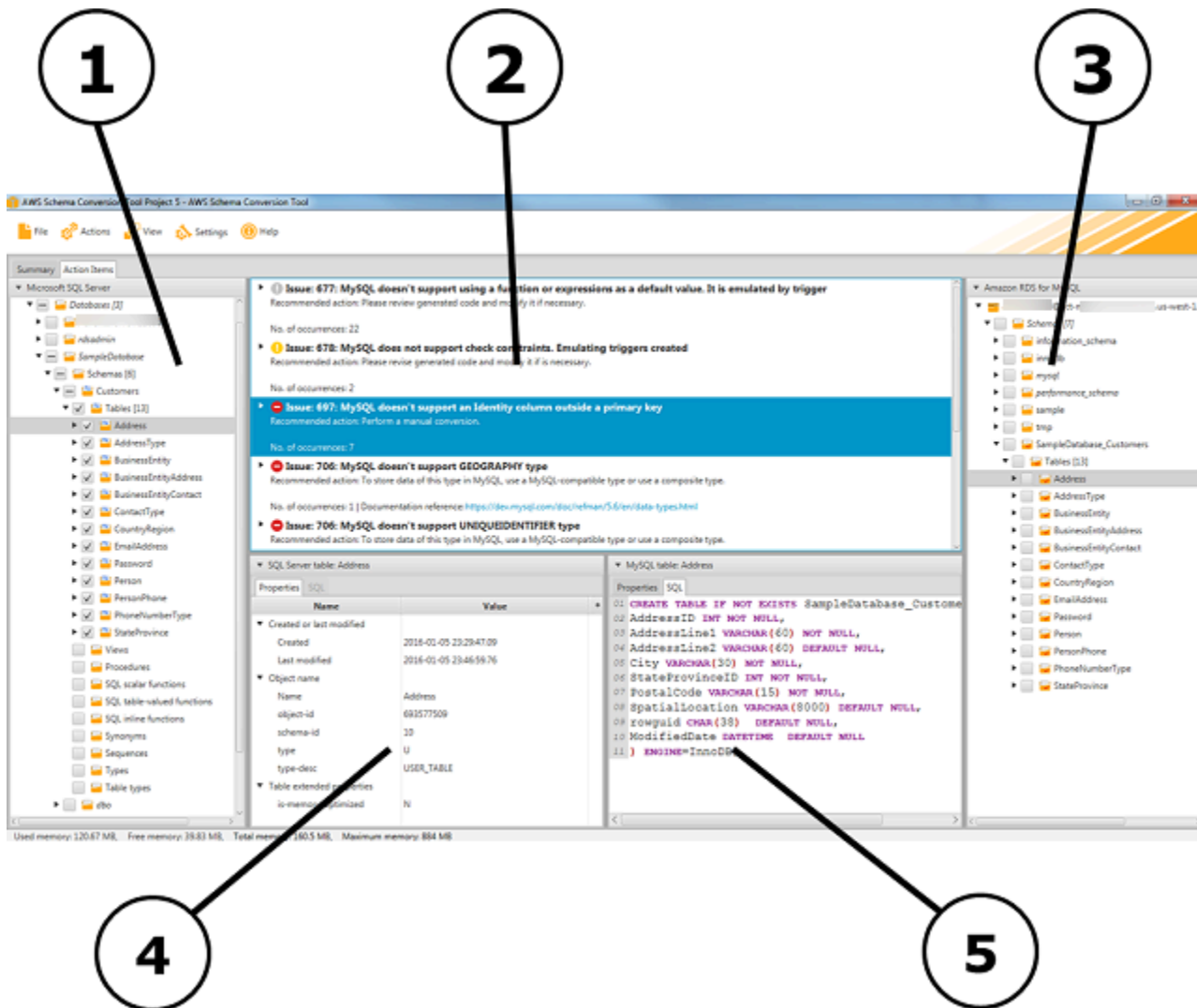
JendelaAWS SCT proyek

Ilustrasi berikut adalah apa yang Anda lihatAWS SCT ketika Anda membuat proyek migrasi skema, dan kemudian mengkonversi skema.

1. Di panel kiri, skema dari basis data dari basis data dari basis data Anda disajikan dalam tampilan pohon. Skema database Anda adalah “lazy loaded.” Dengan kata lain, ketika Anda memilih item

dari tampilan pohon,AWS SCT mendapat dan menampilkan skema saat ini dari database sumber Anda.

2. Di panel tengah atas, item tindakan muncul untuk elemen skema dari mesin database sumber yang tidak dapat dikonversi secara otomatis ke mesin database target.
3. Di panel kanan, skema dari instance DB target Anda disajikan dalam tampilan pohon. Skema database Anda adalah “lazy loaded.” Artinya, pada titik ketika Anda memilih item dari tampilan pohon,AWS SCT mendapat dan menampilkan skema saat ini dari database target Anda.



4. Di panel kiri bawah, ketika Anda memilih elemen skema, properti akan ditampilkan. Ini menggambarkan elemen skema sumber dan perintah SQL untuk membuat elemen dalam database sumber.

5. Di panel kanan bawah, ketika Anda memilih elemen skema, properti akan ditampilkan. Ini menggambarkan elemen skema target dan perintah SQL untuk membuat elemen dalam database target. Anda dapat mengedit perintah SQL ini dan menyimpan perintah yang diperbarui dengan proyek Anda.

Memulai AWS SCT

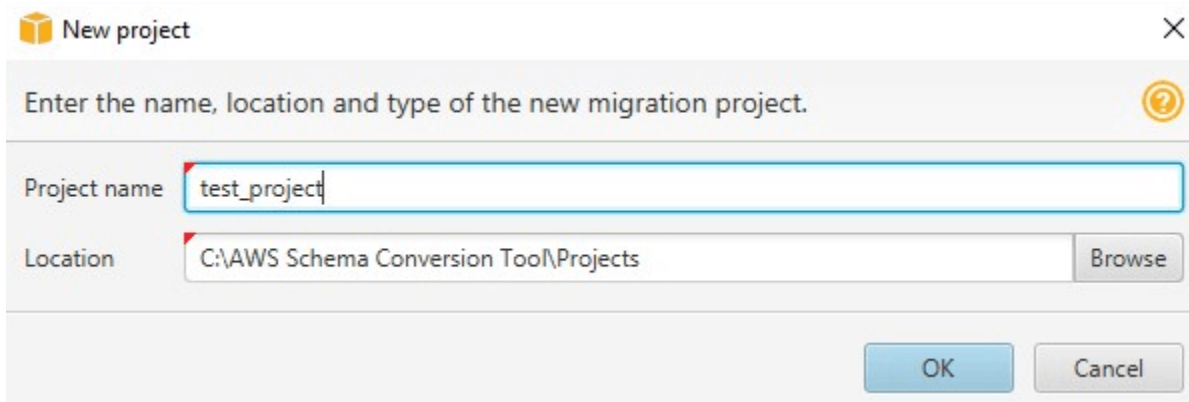
Untuk memulai AWS Schema Conversion Tool, klik dua kali ikon aplikasi.

Membuat AWS SCT proyek

Gunakan prosedur berikut untuk membuat AWS Schema Conversion Tool proyek.

Untuk membuat proyek Anda

1. Memulai AWS Schema Conversion Tool.
2. Pada menu File, pilih Proyek baru. Kotak dialog proyek Baru akan muncul.



3. Memasukkan nama untuk proyek Anda, yang menyimpan secara lokal di komputer Anda.
4. Memasukkan lokasi untuk file proyek lokal Anda.
5. Memilih OKE untuk membuat proyek AWS SCT.
6. Pilih Tambahkan sumber untuk menambahkan database sumber baru ke AWS SCT proyek Anda. Anda dapat menambahkan beberapa database sumber ke AWS SCT proyek Anda.
7. Pilih Tambahkan target untuk menambahkan platform target baru di AWS SCT proyek Anda. Anda dapat menambahkan beberapa platform target ke AWS SCT proyek Anda.
8. Pilih skema database sumber di panel kiri.
9. Di panel kanan, tentukan platform database target untuk skema sumber yang dipilih.

10. Pilih Buat pemetaan. Tombol ini menjadi aktif setelah Anda memilih skema database sumber dan platform database target. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan](#).

Sekarang, AWS SCT proyek Anda sudah diatur. Anda dapat menyimpan proyek Anda, membuat laporan penilaian migrasi database, dan mengonversi skema basis data sumber Anda.

Menggunakan wizard proyek baru di AWS SCT

Anda dapat membuat proyek migrasi database baru menggunakan wizard proyek baru. Wisaya ini membantu Anda dalam menentukan target migrasi dan menghubungkan ke database Anda. Ini memperkirakan seberapa kompleks migrasi untuk semua tujuan target yang didukung. Setelah Anda menjalankan wizard, buat AWS SCT laporan ringkasan untuk migrasi database Anda ke tujuan target yang berbeda. Anda dapat menggunakan laporan ini untuk membandingkan kemungkinan tujuan target dan memilih jalur migrasi yang optimal.

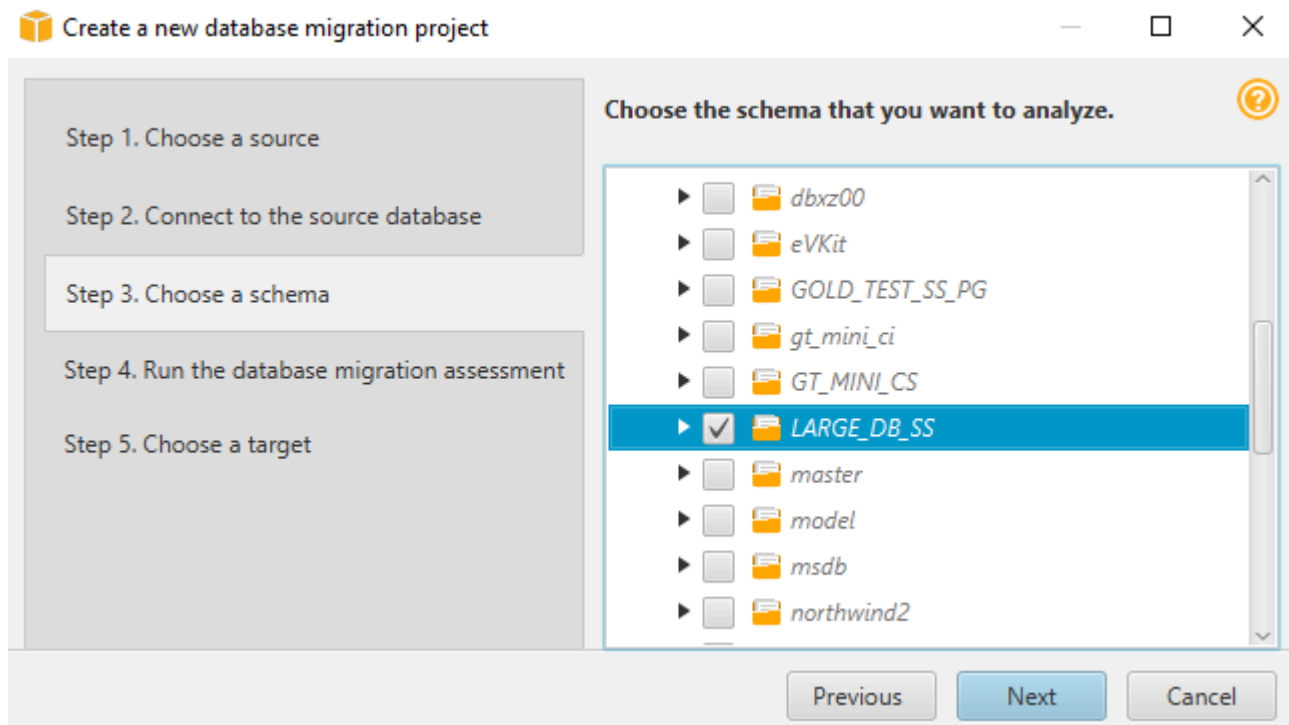
Untuk menjalankan wizard proyek baru

1. Pilih basis data sumber Anda.
 - a. Memulai AWS Schema Conversion Tool.
 - b. Pada menu File, pilih Wisaya proyek baru. Kotak dialog Create a new database migration project terbuka.
 - c. Untuk memasukkan informasi koneksi database sumber, gunakan instruksi berikut:

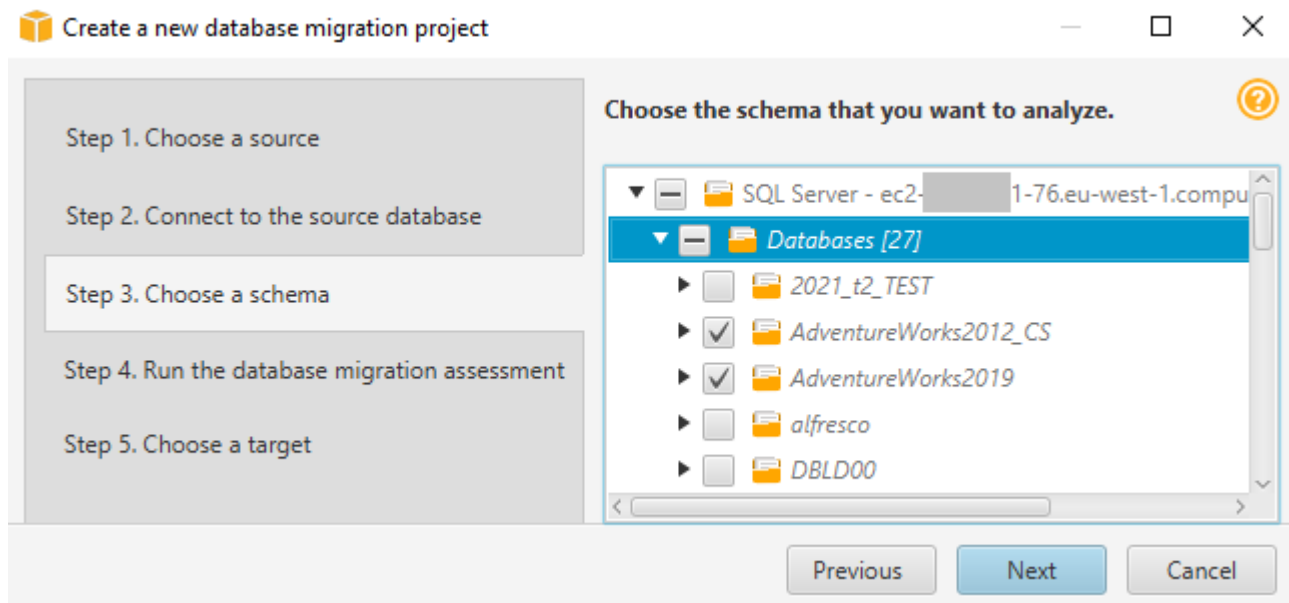
Parameter	Action
Nama proyek	Memasukkan nama untuk proyek Anda, yang menyimpan secara lokal di komputer Anda.
Lokasi	Memasukkan lokasi untuk file proyek lokal Anda.
Jenis sumber	Pilih salah satu opsi berikut: database SQL, database NoSQL, atau ETL. Jika Anda ingin melihat laporan ringkasan yang menyertakan semua tujuan migrasi, pilih database SQL.
Mesin sumber	Pilih mesin basis data sumber Anda.

Parameter	Action
Strategi migrasi	<p>Pilih salah satu opsi berikut:</p> <ul style="list-style-type: none">• Saya ingin mengganti mesin dan mengoptimalkan cloud - Opsi ini mengubah database sumber Anda menjadi mesin database baru.• Saya ingin menyimpan mesin yang sama tetapi mengoptimalkan untuk cloud - Opsi ini membuat mesin database Anda sebagaimana adanya dan memindahkan database dari lokal ke cloud.• Saya ingin melihat laporan gabungan untuk sakelar mesin database dan pengoptimalan untuk cloud - Opsi ini membandingkan kompleksitas migrasi dari semua opsi migrasi yang tersedia. <p>Jika Anda ingin melihat laporan penilaian gabungan yang mencakup semua tujuan migrasi, pilih opsi terakhir.</p>

- d. Pilih Selanjutnya. Halaman Connect to the source database terbuka.
2. Connect ke basis data sumber Anda.
 - a. Berikan informasi koneksi Anda untuk database sumber. Parameter koneksi tergantung pada mesin database sumber Anda. Pastikan pengguna yang Anda gunakan untuk analisis database sumber Anda memiliki izin yang berlaku. Untuk informasi selengkapnya, lihat [Sumber untuk AWS SCT](#).
 - b. Pilih Next (Berikutnya). Halaman Choose a schema terbuka.
 3. Pilih skema basis data Anda.
 - a. Pilih kotak centang untuk nama skema yang ingin Anda nilai dan pilih skema itu sendiri. Nama skema disorot dengan warna biru saat dipilih dan tombol Next tersedia.



- b. Jika Anda ingin menilai beberapa skema database, kemudian pilih kotak centang untuk semua skema dan kemudian pilih node induk. Untuk penilaian yang berhasil, Anda harus memilih node induk. Misalnya, untuk database SQL Server sumber, pilih node Database. Nama node induk disorot dengan warna biru dan tombol Next tersedia.



- c. Pilih Berikutnya. AWS SCT menganalisis skema database sumber Anda dan membuat laporan penilaian migrasi database. Jumlah objek database dalam skema database sumber Anda

mempengaruhi waktu yang diperlukan untuk menjalankan penilaian. Setelah selesai, halaman Jalankan penilaian migrasi database akan terbuka.

4. Jalankan penilaian migrasi database.
 - a. Anda dapat meninjau dan membandingkan laporan penilaian untuk target migrasi yang berbeda atau menyimpan salinan lokal file laporan penilaian untuk analisis lebih lanjut.
 - b. Simpan salinan lokal dari laporan penilaian migrasi database. Pilih Simpan, lalu masukkan path ke folder untuk menyimpan file, dan pilih Simpan. AWS SCT menyimpan file laporan penilaian ke folder yang ditentukan.
 - c. Pilih Selanjutnya. Halaman Pilih target terbuka.
5. Pilih basis data target Anda.
 - a. Untuk mesin Target, pilih mesin database target yang Anda putuskan untuk digunakan berdasarkan laporan penilaian.
 - b. Berikan informasi koneksi Anda untuk database target Anda. Parameter koneksi yang Anda lihat bergantung pada mesin database target yang Anda pilih. Pastikan pengguna yang ditentukan untuk database target memiliki izin yang diperlukan. Untuk informasi selengkapnya tentang izin yang diperlukan, lihat bagian yang menjelaskan izin untuk database target di [Sumber untuk AWS SCT](#) dan [Izin untuk Amazon Redshift sebagai target](#).
 - c. Pilih Selesai. AWS SCT membuat proyek Anda dan menambahkan aturan pemetaan. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan](#).

Sekarang Anda dapat menggunakan AWS SCT proyek untuk mengkonversi objek database sumber Anda.

Menyimpan dan membuka AWS SCT proyek

Gunakan prosedur berikut untuk menyimpan AWS Schema Conversion Tool proyek.

Untuk menyimpan proyek Anda

1. Memulai AWS Schema Conversion Tool.
2. Pada menu File, pilih Simpan proyek.

AWS SCT menyimpan proyek dalam folder, yang Anda tentukan saat Anda membuat proyek.

Gunakan prosedur berikut untuk membuka AWS Schema Conversion Tool proyek yang ada.

Untuk membuka proyek Anda

1. Pada menu File, pilih Open project. Kotak dialog Buka akan muncul.
2. Pilih folder proyek dan kemudian pilih file Windows Script Component (*.sct).
3. AWS SCT membuka proyek Anda tetapi tidak secara otomatis terhubung ke basis data Anda. Pilih Connect ke server di bagian atas pohon skema Anda untuk terhubung ke basis data Anda.

Jika Anda membuka proyek yang disimpan dalam AWS SCT versi 1.0.655 atau sebelumnya, AWS SCT secara otomatis membuat aturan pemetaan untuk semua skema database sumber ke platform database target. Untuk menambahkan platform database target lainnya, hapus aturan pemetaan yang ada, lalu buat aturan pemetaan baru. Untuk informasi lebih lanjut tentang cara membuat aturan pemetaan, lihat [Membuat aturan pemetaan](#).

Menambahkan server database ke AWS SCT proyek

Anda dapat menambahkan beberapa sumber dan target server database untuk AWS Schema Conversion Tool proyek.

Untuk menambahkan server ke proyek Anda

1. Memulai AWS Schema Conversion Tool.
2. Buat proyek baru atau buka proyek yang sudah ada.
3. Pilih Tambahkan sumber dari menu untuk menambahkan database sumber baru.
4. Pilih platform database dan tentukan kredensial koneksi database. Untuk informasi lebih lanjut tentang menghubungkan ke database sumber, lihat [Sumber untuk AWS SCT](#).

Gunakan prosedur berikut untuk terhubung ke basis data Anda.

Untuk terhubung ke basis data Anda

1. Membuka menu konteks (klik kanan) untuk server.

Anda juga dapat memilih Connect ke server di bagian atas pohon skema database Anda.

2. Masukkan kata sandi untuk Connect ke server Anda.

3. Pilih Uji koneksi untuk memverifikasi bahwa Anda AWS SCT dapat terhubung ke basis data Anda.
4. Pilih Connect untuk Connect ke ke basis data Anda.

Gunakan prosedur berikut untuk menghapus AWS SCT server.

Untuk menghapus server database

1. Pilih server basis data yang akan dihapus.
2. Membuka menu konteks (klik kanan), lalu memilih Menghapus dari proyek.

AWS SCT menghapus server database yang dipilih, semua aturan pemetaan, hasil konversi, dan metadata lain yang terkait dengan server ini.

Berjalan AWS SCT dalam mode offline

Anda dapat menjalankan AWS Schema Conversion Tool dalam mode offline. Berikut ini, Anda dapat mempelajari cara bekerja dengan AWS SCT proyek yang ada ketika terputus dari database sumber Anda.

AWS SCT tidak memerlukan koneksi ke database sumber Anda untuk menjalankan operasi berikut:

- Tambahkan aturan pemetaan.
- Buat laporan penilaian migrasi database.
- Mengkonversi skema database dan kode.
- Edit sumber Anda dan kode yang dikonversi.
- Simpan sumber Anda dan kode yang dikonversi sebagai skrip SQL dalam file teks.

Sebelum Anda menggunakan AWS SCT dalam mode offline, sambungkan ke database sumber, muat metadata, dan simpan proyek Anda. Buka proyek ini atau putuskan sambungan dari server database sumber untuk digunakan AWS SCT dalam mode offline.

Menjalankan AWS SCT dalam mode offline

1. Mulai AWS Schema Conversion Tool dan buat proyek baru. Untuk informasi selengkapnya, lihat [Membuat AWS SCT proyek](#).

2. Tambahkan server basis data dan hubungkan ke basis data Anda. Untuk informasi selengkapnya, lihat [Menambahkan server database keAWS SCT proyek](#).
3. Tambahkan server database target atau gunakan platform database target virtual. Untuk informasi selengkapnya, lihat [Menggunakan target virtual](#).
4. Buat aturan pemetaan untuk menentukan platform database target untuk database sumber Anda. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#).
5. Pilih Lihat, lalu pilih Tampilan utama.
6. Di panel kiri yang menampilkan obyek dari basis data Anda, memilih skema. Membuka menu konteks (klik kanan) untuk objek, lalu memilih Memuat skema. Operasi ini memuat semua metadata skema sumber ke dalamAWS SCT proyek Anda.

Operasi Create report dan Convert schema juga memuat semua metadata skema sumber ke dalamAWS SCT proyek Anda. Jika Anda menjalankan salah satu operasi ini dari menu konteks, lewati operasi Load schema.

7. Pada menu File, pilih Simpan proyek untuk menyimpan metadata database sumber dalam proyek Anda.
8. Pilih Putuskan sambungan dari server untuk memutuskan sambungan dari database sumber Anda. Sekarang Anda dapat menggunakanAWS SCT dalam mode offline.

Menggunakan filterAWS SCT pohon

Untuk memigrasi data dari sumber ke target,AWS SCT muat semua metadata dari database sumber dan target ke dalam struktur pohon. Struktur ini munculAWS SCT sebagai tampilan pohon di jendela proyek utama.

Beberapa database dapat memiliki sejumlah besar objek dalam struktur pohon. Anda dapat menggunakan filter pohonAWS SCT untuk mencari objek dalam struktur pohon sumber dan target. Ketika Anda menggunakan filter pohon, Anda tidak mengubah objek yang dikonversi ketika Anda mengkonversi database Anda. Filter hanya mengubah apa yang Anda lihat di pohon.

Filter pohon bekerja dengan objek yangAWS SCT telah dimuat sebelumnya. Dengan kata lain,AWS SCT tidak memuat objek dari database selama pencarian. Pendekatan ini berarti bahwa struktur pohon umumnya mengandung lebih sedikit objek daripada yang ada dalam database.

Untuk filter pohon, ingatlah hal-hal berikut:

- Default filter adalah APAPUN, yang berarti filter menggunakan pencarian nama untuk menemukan objek.
- Bila Anda memilih satu atau lebih jenis objek, Anda hanya melihat jenis-jenis objek di pohon.
- Anda dapat menggunakan masker filter untuk menampilkan berbagai jenis simbol, termasuk Unicode, spasi, dan karakter khusus. Karakter “%” adalah wildcard untuk simbol apa pun.
- Setelah Anda menerapkan filter, hitungan hanya menunjukkan jumlah objek yang disaring.

Untuk membuat filter pohon

1. Buka AWS SCT proyek yang ada.
2. Connect ke basis data yang ingin Anda terapkan pada filter pohon.
3. Pilih ikon filter.




Ikon undo filter berwarna abu-abu karena tidak ada filter yang saat ini diterapkan.

4. Masukkan informasi berikut di Menyaring kotak dialog. Pilihan di kotak dialog berbeda untuk setiap mesin.

AWS SCTopsi filter	Action
Level	<p>Pilih Kategori untuk memfilter objek berdasarkan kategori.</p> <p>Pilih Status untuk menyaring objek berdasarkan status.</p>
Jenis	<p>Untuk Kategori di Level, pilih kategori objek yang disaring. Pilih Setiap dimuat untuk menampilkan objek dari semua kategori.</p> <p>Untuk Status di Level, pilih status objek yang disaring. Anda dapat memilih salah satu opsi berikut:</p> <ul style="list-style-type: none"> • Dikonversi untuk menampilkan semua objek yang dikonversi • Memiliki tindakan untuk menampilkan semua objek yang memiliki masalah konversi • Dienkripsi untuk menampilkan semua objek terenkripsi

AWS SCTopsi filter	Action
Kondisi	<p>Untuk Kategori di Level, pilih kondisi penyaringan antara Suka dan Tidak suka.</p> <p>Untuk Status di Level, opsi kondisi penyaringan tidak tersedia.</p>
Nilai	<p>Untuk Kategori di Level, masukkan Nilai untuk menyaring pohon dengan nilai ini.</p> <p>Gunakan persen (%) sebagai wildcard untuk menampilkan semua objek.</p> <p>Untuk Status di Level, pilih Nilai antara Benar dan Salah.</p>
Dan/Atau	PilihAND atau operatorOR logis untuk menerapkan beberapa klausa filter.

 Filter
×

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

		Level	Type	Condition	Value	And/Or
+	🗑️	Categories	Any loaded	Like	%dbo%	AND
+	🗑️	Categories	Tables	Like	%tmp%	AND
+	🗑️	Statuses	Mapped	Value	True	AND

Add new clause

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

Import
Export
Download template

Reset
Apply
Close

- Pilih Tambahkan klausa baru untuk menambahkan klausa filter tambahan. AWS SCTdapat menerapkan beberapa klausa filter menggunakanAND atau operatorOR logis.

6. Pilih Apply (Terapkan). Setelah Anda memilih Terapkan, ikon undo filter (di samping ikon filter) diaktifkan. Gunakan ikon ini jika Anda ingin menghapus filter yang Anda terapkan.
7. Memilih Tutup untuk menutup kotak dialog.

Ketika Anda memfilter skema yang muncul di pohon, Anda tidak mengubah objek yang dikonversi ketika Anda mengkonversi skema Anda. Filter hanya mengubah apa yang Anda lihat di pohon.

Mengimpor daftar file untuk filter pohon

Anda dapat mengimpor file nilai dipisahkan koma (CSV) dengan pemisah titik koma atau file JSON yang berisi nama atau nilai yang Anda ingin filter pohon untuk digunakan. Buka AWS SCT proyek yang ada, sambungkan ke database untuk menerapkan filter pohon, dan kemudian pilih ikon filter.

Untuk mengunduh contoh file, pilih Unduh template. Masukkan nama file dan pilih Simpan.

Untuk mengunduh pengaturan filter yang ada, pilih Ekspor. Masukkan nama file dan pilih Simpan.

Untuk mengimpor daftar file untuk filter pohon, pilih Impor. Pilih file yang akan diimpor, lalu pilih Buka. Pilih Terapkan, lalu pilih Tutup.

File CSV menggunakan titik koma sebagai pemisah dan memiliki format berikut:

- `object_type` adalah jenis objek yang ingin Anda temukan.
- `database_name` adalah nama database di mana objek ini ada.
- `schema_name` adalah nama skema di mana objek ini ada.
- `object_name` adalah nama objek.
- `import_type` menentukan `include` atau `exclude` ini dari filter.

Gunakan file JSON untuk menjelaskan kasus penyaringan yang kompleks, seperti aturan bersarang.

File JSON memiliki format sebagai berikut:

- `filterGroupType` adalah jenis aturan filter (AND atau operator OR logis) yang berlaku untuk beberapa klausa filter.
- `filterCategory` adalah tingkat filter (Kategori atau Status).
- `name` adalah daftar nama objek yang berlaku untuk filter Kategori.
- `filterCondition` adalah kondisi penyaringan (LIKE atau NOT LIKE) yang berlaku untuk filter Kategori.

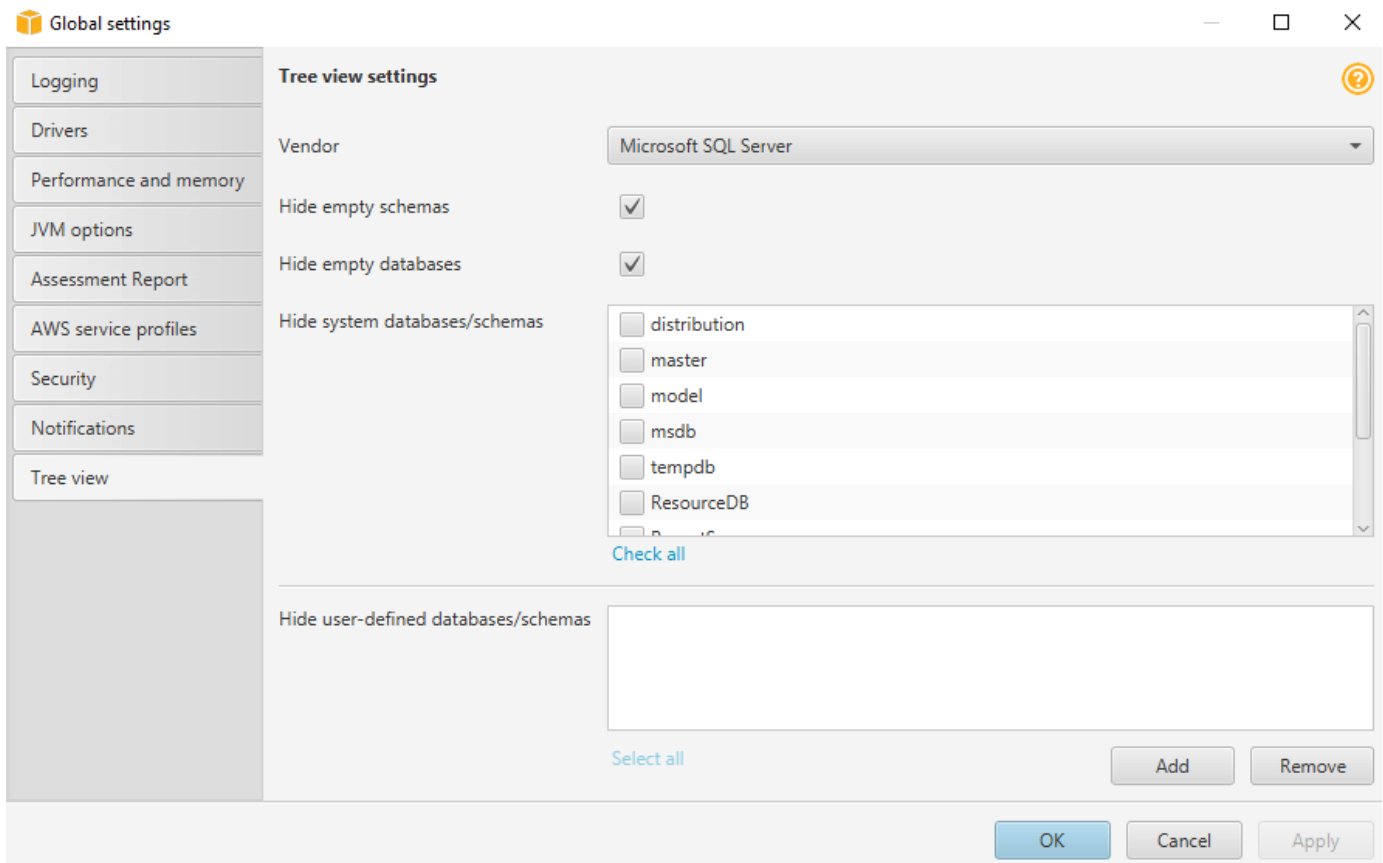
- `transformName` adalah nama status yang berlaku untuk filter Status.
- `value` adalah nilai untuk menyaring pohon oleh.
- `transformValue` adalah nilai filter (TRUEorFALSE) yang berlaku untuk filter Status.

Menyembunyikan skema dalam tampilan AWS SCT pohon

Dengan menggunakan pengaturan tampilan pohon, Anda menentukan skema dan database apa yang ingin Anda lihat dalam tampilan AWS SCT pohon. Anda dapat menyembunyikan skema kosong, database kosong, database sistem, dan database dan skema yang ditentukan pengguna.

Untuk menyembunyikan database dan skema dalam tampilan pohon

1. Buka AWS SCT proyek.
2. Connect ke penyimpanan data yang ingin Anda tampilkan dalam tampilan pohon.
3. Pilih Pengaturan, Pengaturan global, Tampilan pohon.



4. Di bagian Pengaturan tampilan pohon, lakukan hal berikut:

- Untuk Vendor, pilih platform database.
- Pilih Sembunyikan skema kosong untuk menyembunyikan skema kosong untuk platform database yang dipilih.
- Pilih Sembunyikan database kosong untuk menyembunyikan database kosong untuk platform database yang dipilih.
- Untuk Sembunyikan database sistem/skema, pilih database sistem dan skema dengan nama untuk menyembunyikannya.
- Untuk Sembunyikan database/skema yang ditentukan pengguna, masukkan nama database dan skema yang ditentukan pengguna yang ingin Anda sembunyikan, lalu pilih Tambah. Nama-nama tidak peka huruf besar/kecil.

5. Pilih OKE.

Membuat dan meninjau laporan penilaian migrasi database

Laporan penilaian migrasi database merangkum semua item tindakan untuk skema yang tidak dapat dikonversi secara otomatis ke mesin instans Amazon RDS DB target Anda. Laporan ini juga mencakup perkiraan jumlah upaya yang diperlukan untuk menulis kode setara untuk instans DB target Anda.

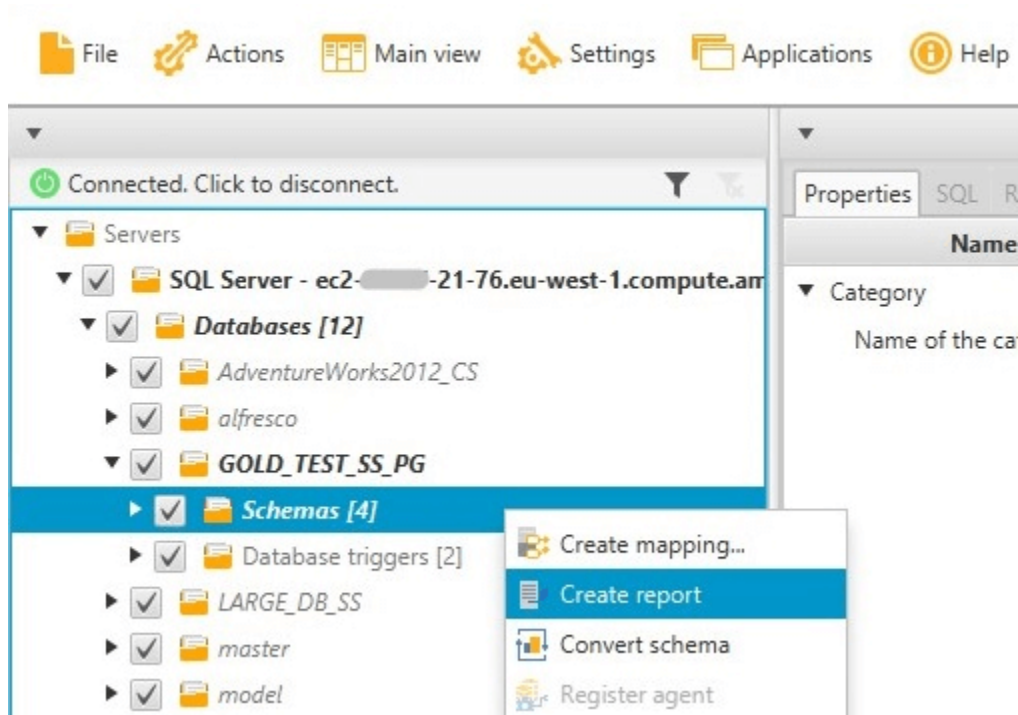
Anda dapat membuat laporan penilaian migrasi database setelah menambahkan basis data sumber dan platform target ke proyek Anda dan menentukan aturan pemetaan.

Membuat dan melihat laporan penilaian migrasi database

1. Pastikan Anda membuat aturan pemetaan untuk skema database sumber untuk membuat laporan penilaian. Untuk informasi selengkapnya, lihat [Menambahkan aturan pemetaan baru](#).
2. Pada menu Lihat, pilih Tampilan utama.
3. Di panel kiri yang menampilkan skema dari basis data dari basis data Anda, memilih obyek skema untuk membuat laporan penilaian.

Pastikan Anda memilih kotak centang untuk semua objek skema yang akan dibuat laporan penilaian.

4. Membuka menu konteks (klik kanan) untuk objek, lalu memilih Membuat laporan.



Tampilan laporan penilaian terbuka.

5. Pilih Item tindakan tab.

Tab Item tindakan menampilkan daftar item yang menggambarkan skema yang tidak dapat dikonversi secara otomatis. Pilih salah satu item tindakan dalam daftar. AWS SCT menyoroti item dari skema Anda bahwa item tindakan berlaku untuk, seperti yang ditunjukkan berikut.

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

Summary 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.
- 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.

Detailed View of Issue 811:

Source: Microsoft SQL Server procedure: POSITION_UPDATE_CASH_CGT_BULK

```

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

Name	Value
Category	Name of the category
	Schemas

6. Pilih tab Ringkasan.

Tab Ringkasan menampilkan informasi ringkasan dari laporan penilaian migrasi database. Ini menunjukkan jumlah item yang dikonversi secara otomatis, dan jumlah item yang tidak dikonversi secara otomatis. Ringkasan ini juga mencakup perkiraan waktu yang diperlukan untuk membuat skema dalam instans DB target Anda yang setara dengan yang ada di database sumber Anda.

Bagian Evaluasi Lisensi dan Support Cloud berisi informasi tentang memindahkan skema database lokal yang ada ke instans DB Amazon RDS yang menjalankan mesin yang sama. Misalnya, jika Anda ingin mengubah jenis lisensi, bagian laporan ini memberi tahu Anda fitur mana dari database Anda saat ini yang akan dihapus.

Contoh ringkasan laporan penilaian ditampilkan sebagai berikut.

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%	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%	0%	100%
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%

- Pilih tab Ringkasan, lalu pilih Simpan ke PDF. Laporan penilaian migrasi database disimpan sebagai file PDF. File PDF berisi informasi ringkasan dan item tindakan.

Anda juga dapat memilih Simpan ke CSV untuk menyimpan laporan sebagai file CSV. Bila Anda memilih opsi ini, AWS SCT buat tiga file CSV. File-file ini berisi informasi berikut:

- Daftar item tindakan konversi dengan tindakan yang direkomendasikan.
- Ringkasan item tindakan konversi dengan perkiraan upaya yang diperlukan untuk mengonversi kejadian item tindakan.
- Ringkasan eksekutif dengan sejumlah item tindakan yang dikategorikan berdasarkan perkiraan waktu untuk mengonversi.

Database objects with conversion actions for Amazon RDS for PostgreSQL

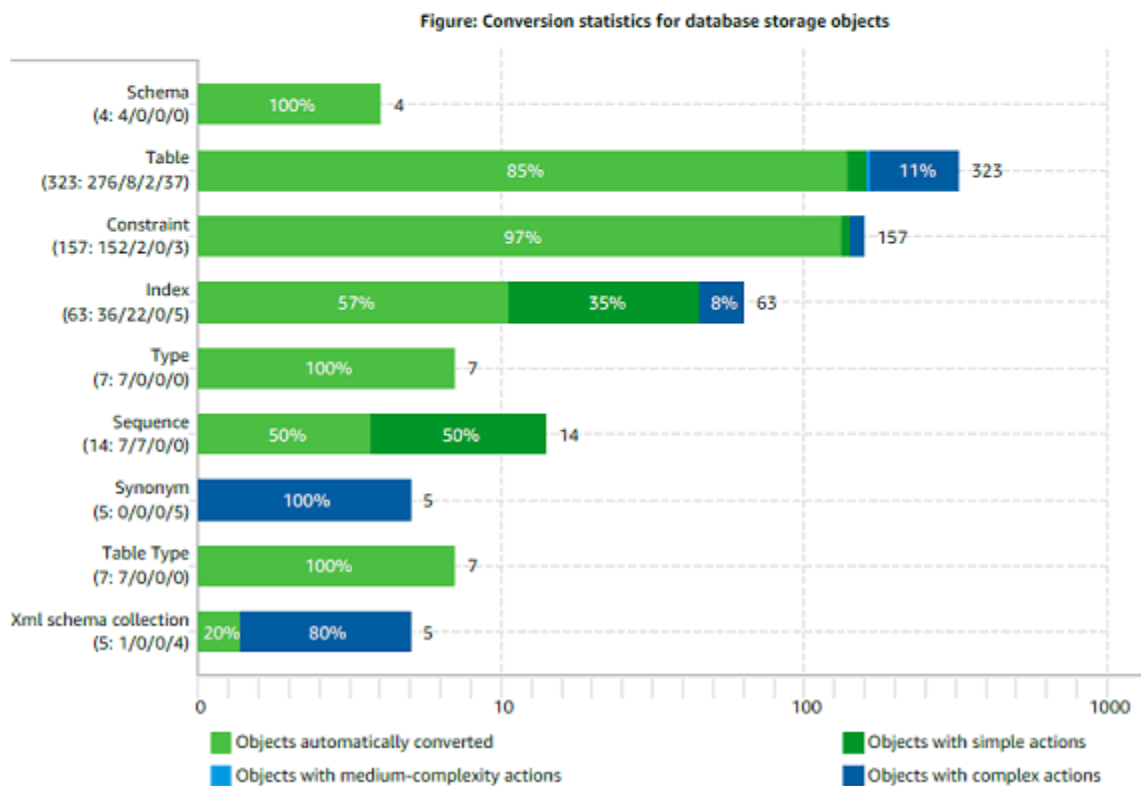
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

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."

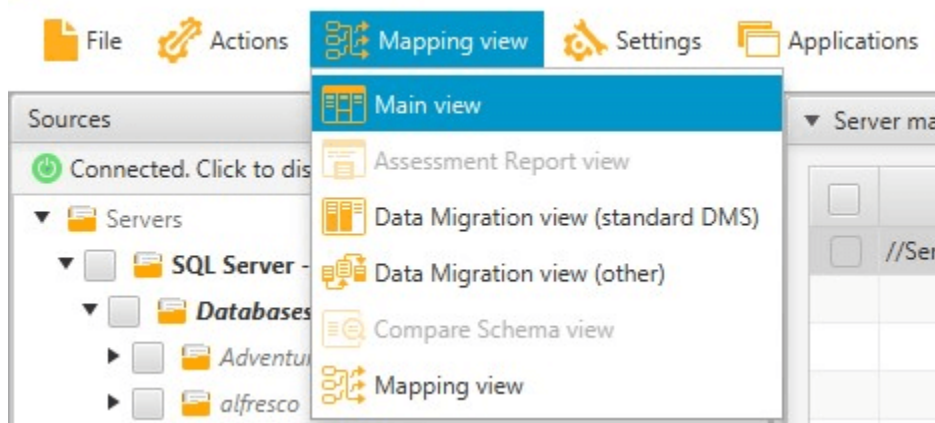


Mengonversi skema Anda

Setelah menambahkan database sumber dan target ke proyek Anda dan membuat aturan pemetaan, Anda dapat mengonversi skema basis data sumber. Gunakan prosedur berikut untuk mengonversi skema.

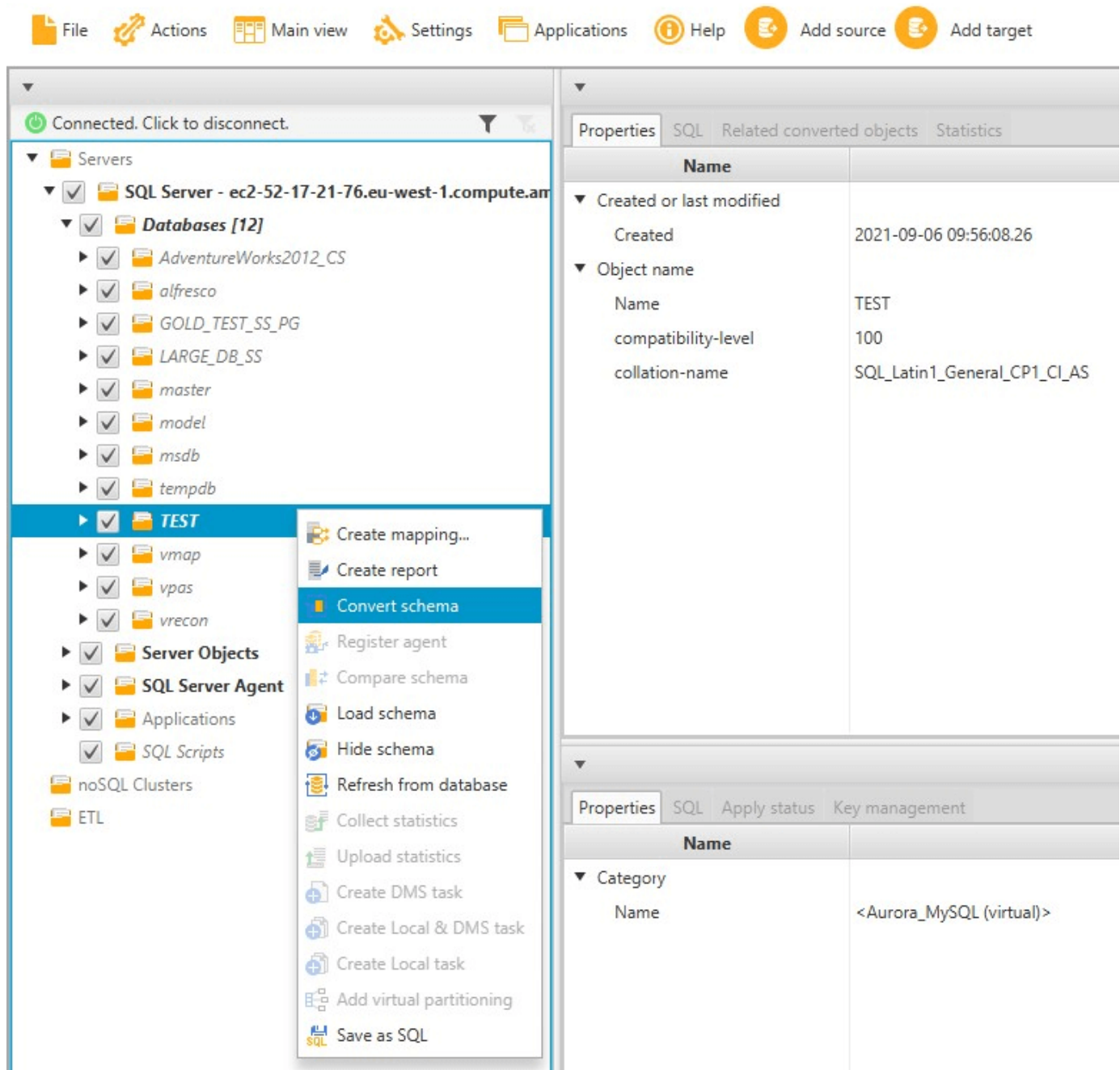
Untuk mengonversi skema Anda

1. Pilih Lihat, lalu pilih Tampilan utama.



2. Di panel kiri yang menampilkan skema dari basis data dari basis data Anda, pilih kotak centang pada nama objek yang akan dikonversi. Selanjutnya, pilih objek ini. AWS SCT menyoroti nama objek dengan warna biru. Membuka menu konteks (klik kanan) untuk objek, lalu memilih Mengubah skema.

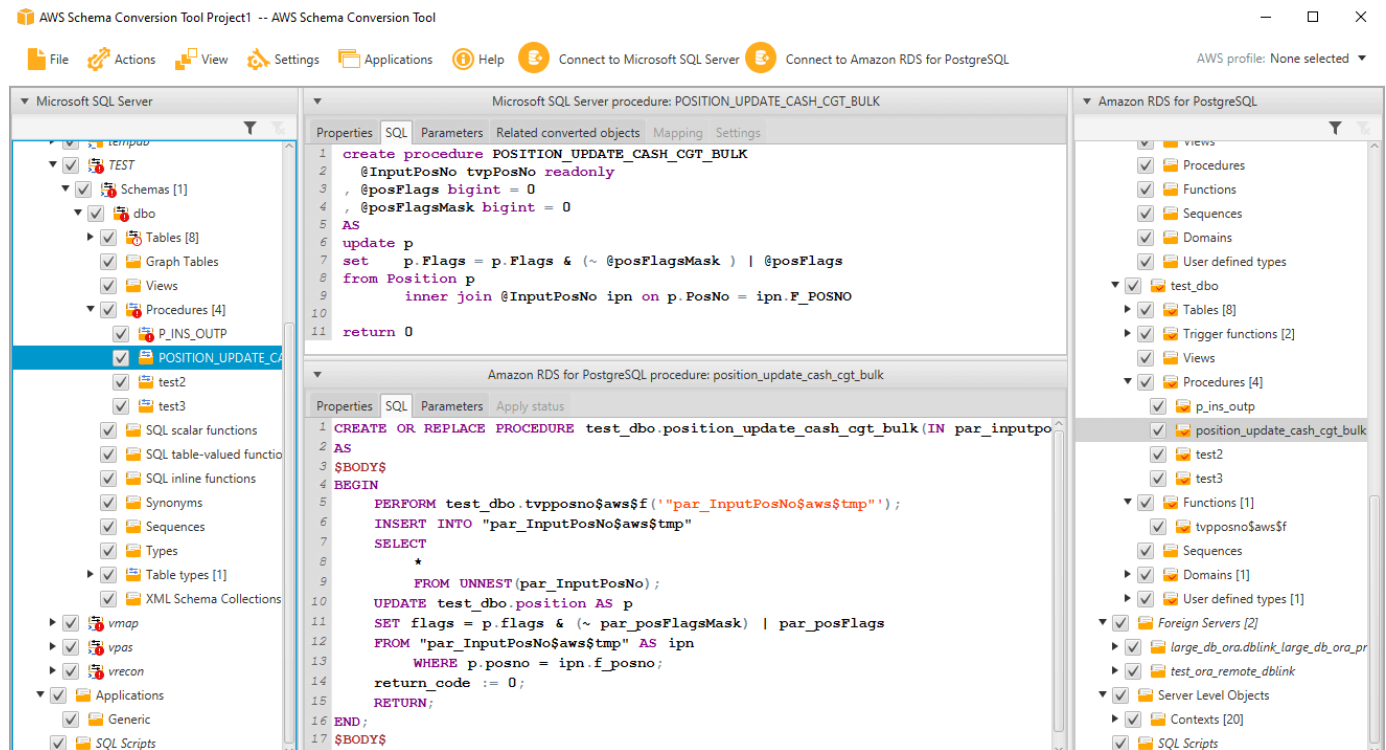
Untuk mengkonversi beberapa objek database, pilih kotak centang untuk semua objek. Selanjutnya, pilih node induk. Misalnya, untuk tabel, node induk adalah Tabel. Pastikan bahwa AWS SCT menyoroti nama node induk berwarna biru. Membuka menu konteks (klik kanan) untuk node induk, lalu memilih Mengubah skema.



3. Ketika AWS SCT selesai mengubah skema, Anda dapat melihat skema yang diusulkan di panel di sebelah kanan proyek Anda.

Pada titik ini, tidak ada skema yang diterapkan pada instance database target Anda. Skema yang direncanakan adalah bagian dari proyek Anda. Jika Anda memilih item skema yang dikonversi, Anda dapat melihat perintah skema yang direncanakan di panel di pusat bawah untuk instance database target Anda.

Anda dapat mengedit skema di jendela ini. Skema yang diedit disimpan sebagai bagian dari proyek Anda dan ditulis ke instance database target saat Anda memilih untuk menerapkan skema yang telah dikonversi.



Menerapkan skema yang dikonversi ke instans DB target Anda

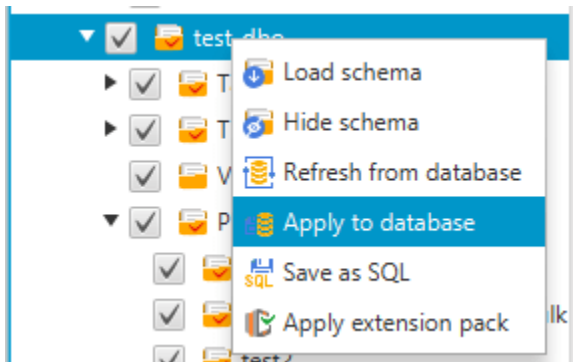
Anda dapat menerapkan skema database yang dikonversi ke instans DB target Anda. Setelah skema diterapkan ke instans DB target, Anda dapat memperbarui skema berdasarkan item tindakan dalam laporan penilaian migrasi database.

⚠ Warning

Prosedur berikut menimpa skema target yang ada. Berhati-hatilah untuk tidak menimpa skema secara tidak sengaja. Berhati-hatilah untuk tidak menimpa skema dalam instans DB target Anda yang telah Anda modifikasi, atau Anda menimpa perubahan tersebut.

Untuk menerapkan skema basis data yang dikonversi ke contoh.

1. Pilih Connect ke server di bagian atas panel kanan proyek Anda untuk terhubung ke database target Anda. Jika Anda terhubung ke database target Anda, maka lewati langkah ini.
2. Pilih elemen skema di panel kanan proyek Anda yang menampilkan skema yang direncanakan untuk instans DB target Anda.
3. Membuka menu konteks (klik kanan) untuk elemen skema, lalu memilih Terapkan ke basis data.



Skema yang dikonversi diterapkan ke instans DB target.

Menyimpan profil AWS layanan di AWS SCT

Anda dapat menyimpan AWS kredensi Anda di AWS SCT. AWS SCT menggunakan kredensi Anda saat Anda menggunakan fitur yang terintegrasi dengan AWS layanan. Misalnya, AWS SCT terintegrasi dengan Amazon S3, AWS Lambda, Amazon Relational Database Service (Amazon RDS), dan AWS Database Migration Service (AWS DMS).

AWS SCT meminta AWS kredensi Anda ketika Anda mengakses fitur yang mengharuskannya. Anda dapat menyimpan kredensial Anda di pengaturan aplikasi global. Saat AWS SCT meminta kredensi Anda, Anda dapat memilih kredensi yang tersimpan.

Anda dapat menyimpan set AWS kredensi yang berbeda dalam pengaturan aplikasi global. Misalnya, Anda dapat menyimpan satu set kredensi yang Anda gunakan dalam skenario pengujian, dan serangkaian kredensi berbeda yang Anda gunakan dalam skenario produksi. Anda juga dapat menyimpan kredensial yang berbeda untuk Wilayah AWS yang berbeda.

Menyimpan AWS kredensi

Gunakan prosedur berikut untuk menyimpan AWS kredensial secara global.

Untuk menyimpan AWS kredensi

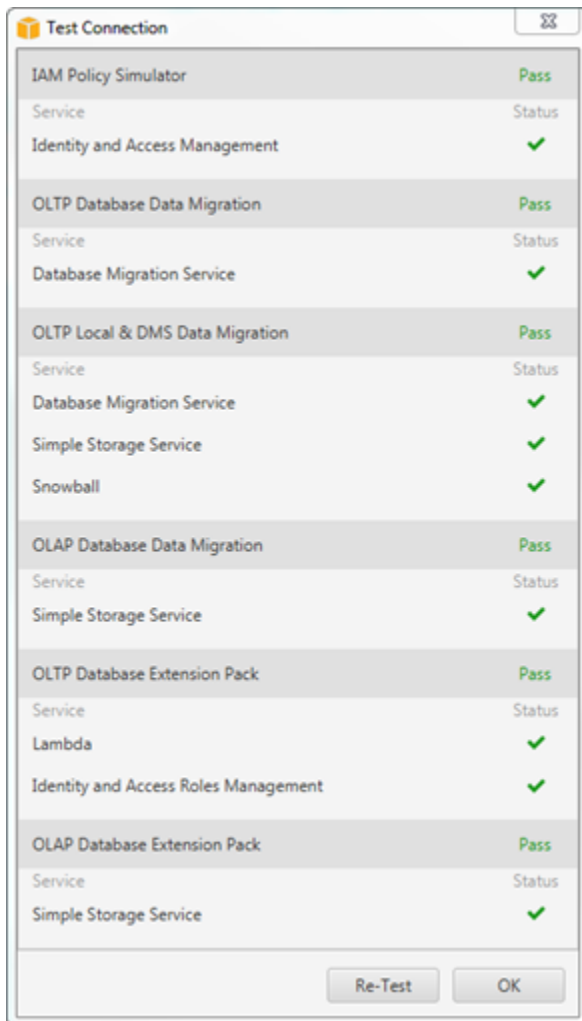
1. Memulai AWS Schema Conversion Tool.
2. Buka menu Pengaturan, lalu pilih Pengaturan global. Kotak dialog parameter Pengaturan global akan muncul.
3. Pilih profil AWS layanan, lalu pilih Tambahkan profil AWS layanan baru.
4. Masukkan AWS informasi Anda sebagai berikut.

AWS SCT pilihan	Action
Nama profil	Masukkan nama untuk profil Anda.
AWS kunci akses	Masukkan kunci AWS akses Anda.
AWS kunci rahasia	Masukkan kunci akses AWS rahasia Anda. Untuk informasi selengkapnya tentang kunci AWS akses, lihat Mengelola kunci akses di Panduan Pengguna IAM.
Wilayah	Pilih Wilayah AWS untuk profil Anda.
Folder bucket Amazon S3	Pilih bucket Amazon S3 untuk profil Anda. Anda perlu menentukan bucket hanya jika Anda menggunakan fitur yang terhubung ke Amazon S3. Untuk informasi lebih lanjut tentang hak istimewa yang diperlukan, lihat Izin untuk menggunakan profil AWS layanan .

Pilih Gunakan FIPS endpoint untuk S3 jika Anda harus sesuai dengan persyaratan keamanan untuk Federal Information Processing Standard (FIPS) jika Anda harus sesuai dengan persyaratan keamanan untuk Federal Information Processing Standard (FIPS). FIPS endpoint tersedia di AWS Wilayah berikut:

- Wilayah Timur AS (Virginia U.)
 - Wilayah US East (Ohio)
 - Wilayah US West (N. California)
 - Wilayah US West (Oregon)
5. Pilih Uji koneksi untuk memverifikasi bahwa kredensi Anda benar dan aktif.

Kotak dialog Koneksi akan muncul. Anda dapat melihat status untuk setiap layanan yang terhubung ke profil Anda. Pass menunjukkan bahwa profil dapat berhasil mengakses layanan.



6. Setelah Anda mengkonfigurasi profil Anda, pilih Simpan untuk menyimpan profil Anda atau Batal untuk membatalkan perubahan Anda.
7. Pilih OK untuk menutup Pengaturan global kotak dialog.

Mengatur profil default untuk sebuah proyek

Anda dapat mengatur profil default untuk sebuah AWS SCT proyek. Melakukan hal ini mengaitkan AWS kredensial yang disimpan dalam profil dengan proyek. Dengan proyek Anda terbuka, gunakan prosedur berikut untuk mengatur profil default.

Untuk mengatur profil default untuk sebuah proyek

1. Mulai AWS Schema Conversion Tool dan buat proyek baru.
2. Pada menu Pengaturan, pilih Pengaturan proyek. Kotak dialog parameter Pengaturan proyek akan muncul.
3. Pilih tab Lingkungan proyek.
4. Pilih Tambahkan profil AWS layanan baru untuk menambahkan profil baru. Kemudian untuk profil AWS layanan, pilih profil yang ingin Anda kaitkan dengan proyek.
5. Pilih OK untuk menutup kotak dialog Project settings. Anda juga dapat memilih Batal untuk membatalkan perubahan Anda.

Izin untuk menggunakan profil AWS layanan

Izin berikut diperlukan untuk mengakses bucket Amazon S3 Anda dari profil AWS layanan Anda:

- `s3:PutObject`— untuk menambahkan objek dalam bucket Amazon S3.
- `s3:DeleteObject`- untuk menghapus objek dan masukkan penanda hapus, yang menjadi versi objek saat ini.
- `s3:ListBucket`— untuk mengembalikan hingga 1.000 objek dari bucket Amazon S3 Anda.
- `s3:GetObject`— untuk mengambil objek dari bucket Amazon S3 Anda.

Contoh kode berikut menunjukkan cara memberikan izin ini kepada pengguna Anda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

```
    }  
  ]  
}
```

Menggunakan AWS Secrets Manager

AWS SCT dapat menggunakan kredensial database yang Anda simpan di AWS Secrets Manager. Anda dapat mengisi semua nilai dalam kotak dialog koneksi database dari Secrets Manager. Untuk menggunakan Secrets Manager, pastikan bahwa Anda menyimpan AWS profil di AWS Schema Conversion Tool.

Untuk informasi lebih lanjut tentang menggunakan AWS Secrets Manager, lihat [Apa itu AWS Secrets Manager?](#) dalam AWS Secrets Manager User Guide. Untuk informasi lebih lanjut tentang menyimpan AWS profil, lihat [Menyimpan profil AWS layanan di AWS SCT](#).

Untuk mengambil kredensi database dari Secrets Manager

1. Mulai AWS Schema Conversion Tool dan buat proyek baru.
2. Pilih Tambahkan sumber atau Tambahkan target untuk menambahkan database baru ke proyek Anda.
3. Pilih platform database dan kemudian pilih Berikutnya.
4. Untuk AWS Secret, pilih rahasia yang ingin Anda gunakan.
5. Pilih Mengisi. Kemudian AWS SCT mengisi semua nilai di kotak dialog koneksi database.
6. Pilih Uji koneksi untuk memverifikasi yang AWS SCT dapat terhubung ke database Anda.
7. Pilih Connect untuk terhubung ke basis data Anda.

AWS SCT mendukung rahasia yang memiliki struktur sebagai berikut.

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

Dalam struktur ini, username dan password nilai-nilai yang diperlukan, dan semua nilai-nilai lainnya adalah opsional. Pastikan bahwa nilai-nilai yang Anda simpan di Secrets Manager mencakup semua kredensi database.

Menyimpan kata sandi basis data

Anda dapat menyimpan kata sandi database atau sertifikat SSL di AWS SCT cache. Untuk menyimpan kata sandi, pilih Simpan Kata Sandi saat Anda membuat sambungan.

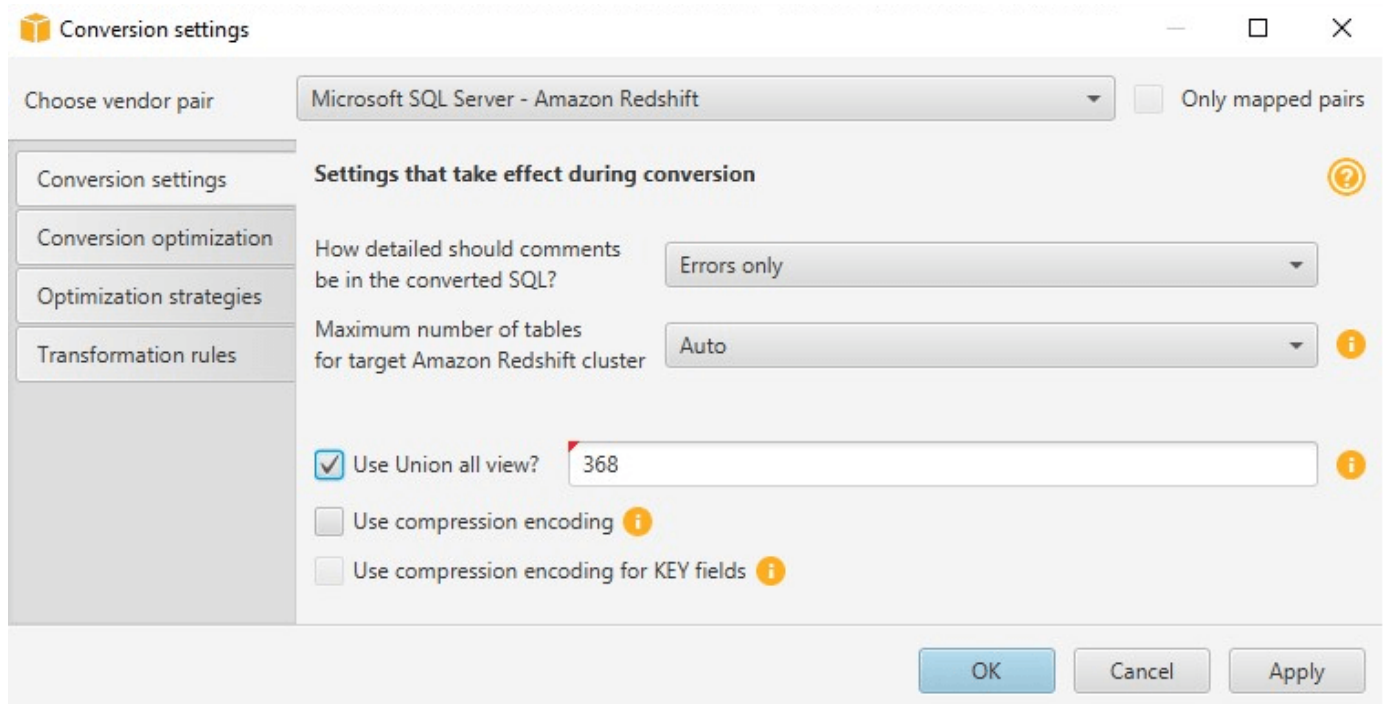
Kata sandi dienkripsi menggunakan token yang dibuat secara acak dalam `seed.dat` file. Kata sandi kemudian disimpan dengan nama pengguna di file cache. Jika Anda kehilangan `seed.dat` file atau menjadi rusak, password database mungkin tidak terenkripsi salah. Dalam kasus ini, koneksi gagal.

Menggunakan tampilan UNION ALL untuk proyek-proyek dengan tabel dipartisi

Jika tabel sumber dipartisi, AWS SCT menciptakan n tabel target, di mana n adalah jumlah partisi pada tabel sumber. AWS SCT menciptakan tampilan UNION ALL di atas tabel target untuk mewakili tabel sumber. Jika Anda menggunakan ekstraktor AWS SCT data untuk memigrasi data Anda, partisi tabel sumber akan diekstraksi dan dimuat secara parallel oleh sub tugas terpisah.

Menggunakan tampilan Union All untuk sebuah proyek

1. Mulai AWS SCT. Buat proyek baru atau buka proyek yang sudah ada AWS SCT.
2. Pada menu Pengaturan, pilih Pengaturan konversi.
3. Pilih sepasang database OLAP dari daftar di bagian atas.
4. Aktifkan Use Union semua tampilan?



- Memilih OK untuk menyimpan pengaturan dan menutup Pengaturan konversi kotak dialog.

Pintasan keyboard untuk AWS SCT

Berikut ini adalah pintasan keyboard yang dapat Anda gunakan AWS SCT.

Pintasan keyboard	Deskripsi
N	Buat proyek baru.
Ctrl+O	Buka proyek yang ada.
Ctrl+S	Simpan proyek terbuka.
Ctrl+W	Buat proyek baru dengan menggunakan wizard.
M	Buat penilaian multiserver baru.
Ctrl+L	Tambahkan basis data sumber baru.
Ctrl+R	Tambahkan basis data target baru.
Ctrl+F4	Tutup proyek terbuka.

Pintasan keyboard	Deskripsi
F1	Buka PanduanAWS SCT Pengguna.

Memulai dengan AWS SCT

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengkonversi skema untuk database sumber. Basis data sumber dapat menjadi mesin yang dikelola sendiri yang berjalan secara On-Premise atau pada instans Amazon EC2. Anda dapat mengonversi skema sumber Anda ke skema untuk database yang didukung yang di-host oleh AWS. Kluster AWS SCT aplikasi menyediakan antarmuka pengguna berbasis proyek.

Hampir semua pekerjaan yang Anda lakukan AWS SCT dimulai dengan langkah-langkah berikut:

1. Instal AWS SCT. Untuk informasi selengkapnya, lihat [Menginstal, memverifikasi, dan memperbarui AWS SCT](#).
2. Pasang AWS SCAgen, jika diperlukan. AWS SCAgen hanya diperlukan untuk skenario migrasi tertentu, seperti antara sumber heterogen dan target. Untuk informasi selengkapnya, lihat [Memigrasi data dari gudang data lokal ke Amazon Redshift](#).
3. Biasakan diri Anda dengan antarmuka pengguna AWS SCT. Untuk informasi selengkapnya, lihat [Menggunakan antarmuka AWS SCT pengguna](#).
4. Buat AWS SCT Proyek. Connect ke basis data sumber dan target. Untuk informasi selengkapnya tentang menghubungkan ke basis data sumber Anda, lihat [Sumber untuk AWS SCT](#).
5. Buat aturan pemetaan. Untuk informasi selengkapnya tentang aturan pemetaan, lihat [Membuat aturan pemetaan di AWS SCT](#).
6. Jalankan dan kemudian tinjau Laporan Penilaian Migrasi Database. Untuk informasi selengkapnya tentang laporan penilaian, lihat [Membuat dan meninjau laporan penilaian migrasi database](#).
7. Konversi skema basis data sumber. Ada beberapa aspek konversi yang perlu Anda ingat, seperti apa yang harus dilakukan dengan item yang tidak dikonversi, dan bagaimana memetakan item yang harus dikonversi dengan cara tertentu. Untuk informasi selengkapnya tentang cara mengubah skema sumber, lihat [Mengonversi skema database menggunakan AWS SCT](#).

Jika Anda mengonversi skema gudang data, ada juga aspek yang perlu Anda pertimbangkan sebelum melakukan konversi. Untuk informasi selengkapnya, lihat [Mengonversi skema gudang data ke Amazon Redshift menggunakan AWS SCT](#).

8. Menerapkan konversi skema ke target Anda. Untuk informasi selengkapnya tentang menerapkan konversi skema sumber, lihat [Menerapkan kode yang dikonversi](#).

9. Anda juga dapat menggunakan AWS SCT untuk mengkonversi SQL disimpan prosedur dan kode aplikasi lainnya. Untuk informasi selengkapnya, lihat [Mengonversi aplikasi SQL menggunakan AWS SCT](#)

Anda juga dapat menggunakan AWS SCT untuk memigrasi data Anda dari basis data sumber ke basis data yang dikelola Amazon. Sebagai contoh, lihat [Memigrasi data dari gudang data lokal ke Amazon Redshift](#).

Sumber untuk AWS SCT

AWS Schema Conversion Tool(AWS SCT) dapat mengkonversi skema dari database sumber berikut dan gudang data ke database target atau gudang data. Untuk informasi tentang izin, koneksi, dan apa yang AWS SCT dapat dikonversi untuk digunakan dengan database target atau gudang data, lihat detail dalam topik berikut.

Informasi enkripsi

[Mengkripsi koneksi Amazon RDS](#)

Sumber database

- [Menggunakan Apache Cassandra sebagai sumber](#)
- [Menggunakan Azure SQL Database sebagai sumber](#)
- [Menggunakan IBM Db2 untuk z/OS sebagai sumber](#)
- [Menggunakan IBM Db2 LUW sebagai sumber](#)
- [Menggunakan MySQL sebagai sumber](#)
- [Menggunakan Oracle Database sebagai sumber](#)
- [Menggunakan PostgreSQL sebagai sumber](#)
- [Menggunakan SAP ASE \(Sybase ASE\) sebagai sumber](#)
- [Menggunakan SQL Server sebagai sumber](#)

Sumber gudang data

- [Menggunakan Amazon Redshift sebagai sumber](#)
- [Menggunakan Azure Synapse Analytics sebagai sumber](#)
- [Menggunakan BigQuery sebagai sumber](#)
- [Menggunakan Greenplum Database sebagai sumber](#)
- [Menggunakan Netezza sebagai sumber](#)
- [Menggunakan Oracle Data Warehouse sebagai sumber](#)
- [Menggunakan Snowflake sebagai sumber](#)
- [Menggunakan SQL Server Data Warehouse sebagai sumber](#)
- [Menggunakan Teradata sebagai sumber](#)

- [Menggunakan Vertica sebagai sumber](#)

Sumber data besar

- [Menggunakan Apache Hadoop sebagai sumber](#)
- [Menggunakan Apache Oozie sebagai sumber](#)

Menkripsi koneksi Amazon RDS dan Amazon Aurora di AWS SCT

Untuk membuka koneksi terenkripsi ke database Amazon RDS atau Amazon Aurora dari aplikasi, Anda perlu mengimpor sertifikat AWS root ke dalam beberapa bentuk penyimpanan kunci. Anda dapat mengunduh sertifikat root dari AWS di [Menggunakan SSL/TLS untuk mengenkripsi koneksi ke instans DB di Panduan](#) Pengguna Amazon RDS.

Dua opsi tersedia, sertifikat root yang berfungsi untuk semua AWS Wilayah dan bundel sertifikat yang berisi sertifikat root lama dan baru.

Bergantung pada yang ingin Anda gunakan, ikuti langkah-langkah di salah satu dari dua prosedur berikut.

Untuk mengimpor sertifikat atau sertifikat ke penyimpanan sistem Windows

1. Unduh sertifikat atau sertifikat dari salah satu sumber berikut:

Untuk informasi tentang mengunduh sertifikat, lihat [Menggunakan SSL/TLS untuk mengenkripsi koneksi ke instans DB](#) di Panduan Pengguna Amazon RDS.

2. Di jendela pencarian Windows Anda, masukkan **Manage computer certificates**. Ketika diminta apakah akan membiarkan aplikasi membuat perubahan pada komputer Anda, pilih Ya.
3. Ketika jendela sertifikat terbuka, jika diperlukan memperluas Sertifikat - Komputer Lokal sehingga Anda dapat melihat daftar sertifikat. Buka menu konteks (klik kanan) untuk Otoritas Sertifikasi Root Terpercaya, lalu pilih Semua Tugas, Impor.
4. Pilih Berikutnya, lalu Jelajahi, dan temukan *.pem file yang Anda unduh di langkah 1. Pilih Buka untuk memilih file sertifikat, pilih Berikutnya, lalu pilih Selesai.

Note

Untuk menemukan file, ubah jenis file di jendela browse ke Semua file (*.*), karena .pem bukan ekstensi sertifikat standar.

5. Di Konsol Manajemen Microsoft, luaskan Sertifikat. Kemudian luaskan Otoritas Sertifikasi Root Tepercaya, pilih Sertifikat, dan temukan sertifikat untuk mengonfirmasi bahwa sertifikat tersebut ada. Nama sertifikat dimulai dengan Amazon RDS.
6. Mulai ulang komputer Anda.

Untuk mengimpor sertifikat atau sertifikat ke Java KeyStore

1. Unduh sertifikat atau sertifikat dari salah satu sumber berikut:

Untuk informasi tentang mengunduh sertifikat, lihat [Menggunakan SSL/TLS untuk mengenkripsi koneksi ke instans DB](#) di Panduan Pengguna Amazon RDS.

2. Jika Anda mengunduh bundel sertifikat, bagi menjadi file sertifikat individual. Untuk melakukannya, tempatkan setiap blok sertifikat, dimulai dengan -----BEGIN CERTIFICATE----- dan diakhiri dengan -----END CERTIFICATE----- ke *.pem file terpisah. Setelah Anda membuat *.pem file terpisah untuk setiap sertifikat, Anda dapat menghapus file bundel sertifikat dengan aman.
3. Buka jendela perintah atau sesi terminal di direktori tempat Anda mengunduh sertifikat, dan jalankan perintah berikut untuk setiap *.pem file yang Anda buat pada langkah sebelumnya.

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

Example

Contoh berikut mengasumsikan bahwa Anda men-download eu-west-1-bundle.pem file.

```
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. Tambahkan keystore sebagai toko kepercayaan di AWS SCT. Untuk melakukannya, dari menu utama pilih Pengaturan, Pengaturan global, Keamanan, Toko kepercayaan, lalu pilih Pilih toko kepercayaan yang ada.

Setelah menambahkan toko kepercayaan, Anda dapat menggunakannya untuk mengkonfigurasi koneksi yang diaktifkan SSL saat Anda membuat AWS SCT sambungan ke database. Dalam dialog AWS SCT Hubungkan ke database, pilih Gunakan SSL dan pilih toko kepercayaan yang dimasukkan sebelumnya.

Menggunakan Apache Cassandra sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengonversi keyspace dari Apache Cassandra ke Amazon DynamoDB.

Menghubungkan ke Apache Cassandra sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Apache Cassandra Anda dengan file. AWS Schema Conversion Tool

Untuk menyambung ke database sumber Apache Cassandra

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Cassandra, lalu pilih Berikutnya.

Kotak dialog Add source muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Rahasia, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi database sumber Apache Cassandra secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama server	Masukkan nama Domain Name Service (DNS) atau alamat IP server database sumber Anda.
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini jika Anda ingin menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none">• Toko kepercayaan: Toko kepercayaan untuk digunakan.• Toko kunci: Toko kunci untuk digunakan.

Parameter	Action
Simpan kata sandi	AWS SCTmenciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.

5. Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
6. Pilih Hubungkan untuk terhubung ke database sumber Anda.

Menggunakan Apache Hadoop sebagai sumber untuk AWS SCT

Anda dapat menggunakan antarmuka baris AWS SCT perintah (CLI) untuk bermigrasi dari Apache Hadoop ke Amazon EMR. AWS SCTmenggunakan bucket Amazon S3 Anda sebagai penyimpanan sementara untuk data Anda selama migrasi.

AWS SCTmendukung sebagai sumber Apache Hadoop versi 2.2.0 dan lebih tinggi. Juga, AWS SCT mendukung Apache Hive versi 0.13.0 dan lebih tinggi.

AWS SCTmendukung sebagai target Amazon EMR versi 6.3.0 dan yang lebih tinggi. Juga, AWS SCT mendukung sebagai target Apache Hadoop versi 2.6.0 dan lebih tinggi, dan Apache Hive versi 0.13.0 dan lebih tinggi.

Topik

- [Prasyarat untuk menggunakan Apache Hadoop sebagai sumber](#)
- [Izin untuk menggunakan Hive sebagai sumber](#)
- [Izin untuk menggunakan HDFS sebagai sumber](#)
- [Izin untuk menggunakan HDFS sebagai target](#)
- [Menghubungkan ke Apache Hadoop sebagai sumber](#)
- [Menghubungkan ke layanan sumber Hive dan HDFS Anda](#)
- [Menghubungkan ke Amazon EMR sebagai target](#)

Prasyarat untuk menggunakan Apache Hadoop sebagai sumber

Prasyarat berikut diperlukan untuk terhubung ke Apache Hadoop dengan CLI. AWS SCT

- Buat bucket Amazon S3 untuk menyimpan data selama migrasi. Anda kemudian dapat menyalin data ke Amazon EMR HDFS atau menggunakan Amazon S3 sebagai repositori data untuk beban kerja Hadoop Anda. Untuk informasi selengkapnya, lihat [Membuat bucket](#) di Panduan Pengguna Amazon S3.
- Buat peran AWS Identity and Access Management (IAM) dengan `AmazonS3FullAccess` kebijakan. AWS SCT menggunakan peran IAM ini untuk mengakses bucket Amazon S3 Anda.
- Catat kunci AWS rahasia dan kunci akses AWS rahasia Anda. Untuk informasi selengkapnya tentang kunci AWS akses, lihat [Mengelola kunci akses](#) di Panduan Pengguna IAM.
- Buat dan konfigurasi kluster Amazon EMR target. Untuk informasi selengkapnya, lihat [Memulai Amazon EMR](#) di Panduan Manajemen Amazon EMR.
- Instal `distcp` utilitas pada cluster Apache Hadoop sumber Anda. Juga, instal `s3-dist-cp` utilitas pada kluster Amazon EMR target Anda. Pastikan bahwa pengguna database Anda memiliki izin untuk menjalankan utilitas ini.
- Konfigurasi `core-site.xml` file di cluster Hadoop sumber Anda untuk menggunakan protokol `s3a`. Untuk melakukannya, atur `fs.s3a.aws.credentials.provider` parameter ke salah satu nilai berikut.
 - `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`

Anda dapat menambahkan contoh kode berikut ke dalam `core-site.xml` file.

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

Contoh sebelumnya menunjukkan salah satu dari empat pilihan dari daftar sebelumnya pilihan. Jika Anda tidak mengatur `fs.s3a.aws.credentials.provider` parameter dalam `core-site.xml` file, AWS SCT pilih penyedia secara otomatis.

Izin untuk menggunakan Hive sebagai sumber

Izin yang diperlukan untuk pengguna sumber Hive adalah sebagai berikut:

- READ akses ke folder data sumber dan ke bucket Amazon S3 sumber
- READ+WRITE akses ke bucket Amazon S3 menengah dan target

Untuk meningkatkan kecepatan migrasi, sebaiknya jalankan pemadatan untuk tab sumber transaksional Asam.

Izin yang diperlukan untuk pengguna target Amazon EMR Hive adalah sebagai berikut:

- READ akses ke bucket Amazon S3 target
- READ+WRITE akses ke bucket Amazon S3 perantara
- READ+WRITE akses ke folder target HDFS

Izin untuk menggunakan HDFS sebagai sumber

Izin yang diperlukan untuk HDFS sebagai sumber adalah sebagai berikut:

- EXECUTE untuk NameNode
- EXECUTE+READ untuk semua folder sumber dan file yang Anda sertakan dalam proyek migrasi
- READ+WRITE untuk tmp direktori dalam menjalankan pekerjaan Spark dan menyimpan file sebelum migrasi ke Amazon S3 NameNode

Dalam HDFS, semua operasi memerlukan akses traversal. Akses traversal menuntut EXECUTE izin pada semua komponen jalur yang ada, kecuali untuk komponen jalur akhir. Misalnya, untuk setiap operasi yang mengakses `/foo/bar/baz`, pengguna Anda harus memiliki EXECUTE izin `/`, `/foo`, dan `foo/bar`.

Contoh kode berikut menunjukkan cara memberikan EXECUTE+READ izin untuk folder dan file sumber Anda, dan READ+WRITE izin untuk direktori `tmp`

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

Izin untuk menggunakan HDFS sebagai target

Izin yang diperlukan untuk Amazon EMR HDFS sebagai target adalah sebagai berikut:

- EXECUTE untuk NameNode klaster Amazon EMR target
- READ+WRITE untuk folder target HDFS tempat Anda akan menyimpan data setelah migrasi

Menghubungkan ke Apache Hadoop sebagai sumber

Anda dapat menggunakan Apache Hadoop sebagai sumber dalam AWS SCT versi 1.0.670 atau lebih tinggi. Anda dapat memigrasi klaster Hadoop ke Amazon EMR hanya di antarmuka baris AWS SCT perintah (CLI). Sebelum Anda mulai, biasakan diri Anda dengan antarmuka baris perintah dari AWS SCT Untuk informasi selengkapnya, lihat [AWS SCT Referensi CLI](#).

Untuk terhubung ke Apache Hadoop di CLI AWS SCT

1. Buat skrip AWS SCT CLI baru atau edit template skenario yang ada. Misalnya, Anda dapat mengunduh dan mengedit `HadoopMigrationTemplate.scts` template. Untuk informasi selengkapnya, lihat [Mendapatkan skenario CLI](#).
2. Konfigurasi pengaturan AWS SCT aplikasi seperti lokasi driver dan folder log.

Unduh driver JDBC yang diperlukan dan tentukan lokasi tempat Anda menyimpan file. Untuk informasi selengkapnya, lihat [Mengunduh driver database yang diperlukan](#).

Contoh kode berikut menunjukkan cara menambahkan path ke driver Apache Hive. Setelah Anda menjalankan contoh kode ini, AWS SCT menyimpan file log dalam `c:\sct` folder.

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

Anda dapat menggunakan contoh ini dan contoh berikut di Windows.

3. Buat AWS SCT proyek baru.

Contoh kode berikut membuat `hadoop_emr` proyek dalam `c:\sct` folder.

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

4. Tambahkan cluster Hadoop sumber Anda ke proyek.

Gunakan `AddSourceCluster` perintah untuk terhubung ke cluster Hadoop sumber. Pastikan bahwa Anda memberikan nilai untuk parameter wajib berikut: `name`, `host`, `port`, dan `user`. Parameter lainnya bersifat opsional.

Contoh kode berikut menambahkan sumber Hadoop cluster. Contoh ini ditetapkan `HADOOP_SOURCE` sebagai nama cluster sumber. Gunakan nama objek ini untuk menambahkan layanan Hive dan HDFS ke proyek dan membuat aturan pemetaan.

```
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'
/
```

Dalam contoh sebelumnya, ganti *hadoop_address* dengan *alamat* IP cluster Hadoop Anda. Jika diperlukan, konfigurasi nilai opsi port. Selanjutnya, ganti *hadoop_user* dan *hadoop_password* dengan *nama pengguna Hadoop Anda dan kata sandi* untuk pengguna ini. Untuk *path\ name*, masukkan nama dan path ke file PEM untuk cluster Hadoop sumber Anda.

5. Simpan skrip CLI Anda. Selanjutnya, tambahkan informasi koneksi untuk layanan Hive dan HDFS Anda.

Menghubungkan ke layanan sumber Hive dan HDFS Anda

Anda dapat terhubung ke sumber Hive dan HDFS layanan dengan CLI. AWS SCT Untuk terhubung ke Apache Hive, gunakan driver Hive JDBC versi 2.3.4 atau lebih tinggi. Untuk informasi selengkapnya, lihat [Mengunduh driver database yang diperlukan](#).

AWS SCTterhubung ke Apache Hive dengan pengguna hadoop cluster. Untuk melakukannya, gunakan `AddSourceClusterHive` dan `AddSourceClusterHDFS` perintah. Anda dapat menggunakan salah satu pendekatan berikut.

- Buat terowongan SSH baru.

Untuk `createTunnel`, masukkan **true**. Untuk `host`, masukkan alamat IP internal sumber Hive atau layanan HDFS Anda. Untuk `port`, masukkan port layanan layanan Hive atau HDFS Anda.

Selanjutnya, masukkan kredensi Hive atau HDFS Anda untuk `dan. user password` Untuk informasi selengkapnya tentang terowongan SSH, lihat [Menyiapkan terowongan SSH ke node utama menggunakan penerusan port lokal](#) di Panduan Manajemen Amazon EMR.

- Gunakan terowongan SSH yang ada.

Untuk `host`, masukkan **localhost**. Untuk `port`, masukkan port lokal dari parameter terowongan SSH.

- Hubungkan ke layanan Hive dan HDFS Anda secara langsung.

Untuk `host`, masukkan alamat IP atau nama host dari sumber Hive atau layanan HDFS Anda. Untuk `port`, masukkan port layanan layanan Hive atau HDFS Anda. Selanjutnya, masukkan kredensi Hive atau HDFS Anda untuk `dan. user password`

Untuk terhubung ke Hive dan HDFS di CLI AWS SCT

1. Buka skrip CLI Anda yang mencakup informasi koneksi untuk kluster Hadoop sumber Anda. Pastikan Anda menggunakan nama kluster Hadoop yang Anda tentukan pada langkah sebelumnya.
2. Tambahkan layanan sumber Hive Anda ke proyek.

Gunakan `AddSourceClusterHive` perintah untuk menghubungkan layanan sumber Hive. Pastikan bahwa Anda memberikan nilai untuk parameter wajib berikut: `user`, `password`, `cluster`, `name`, dan `port`. Parameter lainnya bersifat opsional.

Contoh kode berikut membuat terowongan AWS SCT untuk bekerja dengan layanan Hive Anda. Layanan sumber Hive ini berjalan pada PC yang sama dengan. AWS SCT Contoh ini menggunakan cluster HADOOP_SOURCE sumber dari contoh sebelumnya.

```
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'
/
```

Contoh kode berikut terhubung ke layanan Hive Anda tanpa terowongan.

```
AddSourceClusterHive
  -cluster: 'HADOOP_SOURCE'
  -name: 'HIVE_SOURCE'
  -host: 'hive_address'
  -port: 'hive_port'
  -user: 'hive_user'
  -password: 'hive_password'
/
```

Dalam contoh sebelumnya, ganti *hive_user* dan *hive_password* dengan nama pengguna Hive Anda dan kata sandi untuk pengguna ini.

Selanjutnya, ganti *hive_address* dan *hive_port* dengan alamat NameNode IP dan port cluster Hadoop sumber Anda.

Untuk *hive_remote_address*, Anda dapat menggunakan nilai default 127.0.0.1 atau alamat NameNode IP dari layanan sumber Hive Anda.

3. Tambahkan layanan HDFS sumber Anda ke proyek.

Gunakan `AddSourceClusterHDFS` perintah untuk menghubungkan layanan sumber HDFS. Pastikan bahwa Anda memberikan nilai untuk parameter wajib berikut: `user`, `password`, `cluster`, `name`, dan `port`. Parameter lainnya bersifat opsional.

Pastikan bahwa pengguna Anda memiliki izin yang diperlukan untuk memigrasi data dari layanan sumber HDFS Anda. Untuk informasi selengkapnya, lihat [Izin untuk menggunakan Hive sebagai sumber](#).

Contoh kode berikut membuat terowongan AWS SCT untuk bekerja dengan layanan Apache HDFS Anda. Contoh ini menggunakan cluster `HADOOP_SOURCE` sumber yang Anda buat sebelumnya.

```
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'
```

/

Kode berikut terhubung ke layanan Apache HDFS Anda tanpa terowongan.

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

/

Dalam contoh sebelumnya, ganti *hdfs_user* dan *hdfs_password* dengan nama pengguna HDFS Anda dan kata sandi untuk pengguna ini.

Selanjutnya, ganti *hdfs_address* dan *hdfs_port* dengan alamat NameNode IP dan *port cluster* Hadoop sumber Anda.

Untuk *hdfs_remote_address*, Anda dapat menggunakan nilai default 127.0.0.1 atau alamat NameNode IP dari layanan sumber Hive Anda.

4. Simpan skrip CLI Anda. Selanjutnya, tambahkan informasi koneksi untuk kluster Amazon EMR target Anda, dan perintah migrasi.

Menghubungkan ke Amazon EMR sebagai target

Anda dapat terhubung ke kluster Amazon EMR target Anda dengan CLI. AWS SCT Untuk melakukannya, Anda mengotorisasi lalu lintas masuk dan menggunakan SSH. Dalam hal ini, AWS SCT memiliki semua izin yang diperlukan untuk bekerja dengan kluster Amazon EMR Anda. Untuk informasi selengkapnya, lihat [Sebelum Anda terhubung](#) dan [Terhubung ke node utama menggunakan SSH](#) di Amazon EMR Management Guide.

AWS SCTterhubung ke Amazon EMR Hive dengan pengguna hadoop kluster. Untuk terhubung ke Amazon EMR Hive, gunakan driver Hive JDBC versi 2.6.2.1002 atau yang lebih tinggi. Untuk informasi selengkapnya, lihat [Mengunduh driver database yang diperlukan](#).

Untuk terhubung ke Amazon EMR di CLI AWS SCT

1. Buka skrip CLI Anda yang mencakup informasi koneksi untuk kluster Hadoop sumber Anda. Tambahkan kredensi Amazon EMR target ke dalam file ini.
2. Tambahkan kluster Amazon EMR target Anda ke proyek.

Contoh kode berikut menambahkan kluster Amazon EMR target. Contoh ini ditetapkan HADOOP_TARGET sebagai nama cluster target. Gunakan nama objek ini untuk menambahkan layanan Hive dan HDFS dan Amazon S3, folder bucket ke proyek dan membuat aturan pemetaan.

```
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 '  
/
```

Dalam contoh sebelumnya, masukkan nama AWS sumber daya Anda dan informasi koneksi Amazon EMR. Ini termasuk alamat IP kluster Amazon EMR Anda, kunci AWS akses, kunci akses AWS rahasia, dan bucket Amazon S3 Anda. Jika diperlukan, konfigurasi nilai variabel port. Selanjutnya, ganti *emr_user* dan *emr_password* dengan nama pengguna Amazon EMR Anda dan kata sandi untuk pengguna ini. Untuk *path\ name*, masukkan nama dan path ke file PEM untuk kluster Amazon EMR target Anda. Untuk informasi selengkapnya, lihat [Mengunduh File PEM untuk Akses Cluster EMR](#).

3. Tambahkan bucket Amazon S3 target Anda ke proyek.

Contoh kode berikut menambahkan bucket Amazon S3 target. Contoh ini menggunakan HADOOP_TARGET cluster yang Anda buat sebelumnya.

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

Pada contoh sebelumnya, masukkan kunci akses, kunci AWS akses AWS rahasia, dan bucket Amazon S3 Anda.

4. Tambahkan layanan Hive target Anda ke proyek.

Contoh kode berikut membuat terowongan AWS SCT untuk bekerja dengan layanan sarang target Anda. Contoh ini menggunakan kluster HADOOP_TARGET target yang Anda buat sebelumnya.


```
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'
/
```

Pada contoh sebelumnya, ganti *hive_user* dan *hive_password* dengan nama pengguna *Hive* Anda dan kata sandi untuk pengguna ini.

Selanjutnya, ganti *hive_address* dengan nilai default `127.0.0.1` atau dengan alamat NameNode IP layanan Hive target Anda. Selanjutnya, ganti *hive_port* dengan *port* layanan Hive target Anda.

5. Tambahkan layanan HDFS target Anda ke proyek.

Contoh kode berikut membuat terowongan AWS SCT untuk bekerja dengan layanan Apache HDFS Anda. Contoh ini menggunakan kluster `HADOOP_TARGET` target yang Anda buat sebelumnya.

```
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'
/
```

Pada contoh sebelumnya, ganti *hdfs_user* dan *hdfs_password* dengan nama pengguna *HDFS* Anda dan kata sandi untuk pengguna ini.

Selanjutnya, ganti `hdfs_address` dan `hdfs_port` dengan alamat IP pribadi dan port dari layanan HDFS target Anda. NameNode

6. Simpan skrip CLI Anda. Selanjutnya, tambahkan aturan pemetaan dan perintah migrasi. Untuk informasi selengkapnya, lihat [Memigrasi Apache Hadoop ke Amazon EMR](#).

Menggunakan Apache Oozie sebagai sumber untuk AWS SCT

Anda dapat menggunakan antarmuka baris AWS SCT perintah (CLI) untuk mengubah alur kerja Apache Oozie ke AWS Step Functions. Setelah Anda memigrasi beban kerja Apache Hadoop ke Amazon EMR, Anda dapat menggunakan layanan native di untuk mengatur pekerjaan Anda. AWS Cloud Untuk informasi selengkapnya, lihat [Menggunakan Apache Hadoop sebagai sumber](#).

AWS SCT mengubah alur kerja Oozie Anda ke AWS Step Functions dan menggunakan AWS Lambda untuk meniru fitur yang tidak mendukung. AWS Step Functions Juga, AWS SCT mengkonversi properti pekerjaan Oozie Anda untuk AWS Systems Manager

Untuk mengonversi alur kerja Apache Oozie, pastikan Anda menggunakan AWS SCT versi 1.0.671 atau yang lebih tinggi. Juga, biasakan diri Anda dengan antarmuka baris perintah dari AWS SCT Untuk informasi selengkapnya, lihat [AWS SCT Referensi CLI](#).

Prasyarat untuk menggunakan Apache Oozie sebagai sumber

Prasyarat berikut diperlukan untuk terhubung ke Apache Oozie dengan CLI. AWS SCT

- Buat bucket Amazon S3 untuk menyimpan definisi mesin status. Anda dapat menggunakan definisi ini untuk mengkonfigurasi mesin negara Anda. Untuk informasi selengkapnya, lihat [Membuat bucket](#) di Panduan Pengguna Amazon S3.
- Buat peran AWS Identity and Access Management (IAM) dengan `AmazonS3FullAccess` kebijakan. AWS SCT menggunakan peran IAM ini untuk mengakses bucket Amazon S3 Anda.
- Catat kunci AWS rahasia dan kunci akses AWS rahasia Anda. Untuk informasi selengkapnya tentang kunci AWS akses, lihat [Mengelola kunci akses](#) di Panduan Pengguna IAM.
- Simpan AWS kredensi Anda dan informasi tentang bucket Amazon S3 Anda di profil AWS layanan di pengaturan aplikasi global. Kemudian, AWS SCT gunakan profil AWS layanan ini untuk bekerja dengan AWS sumber daya Anda. Untuk informasi selengkapnya, lihat [Menyimpan profil AWS layanan di AWS SCT](#).

Untuk bekerja dengan alur kerja Apache Oozie sumber Anda, AWS SCT memerlukan struktur spesifik file sumber Anda. Setiap folder aplikasi Anda harus menyertakan `job.properties` file tersebut. File ini mencakup pasangan kunci-nilai properti pekerjaan Anda. Selain itu, setiap folder aplikasi Anda harus menyertakan `workflow.xml` file tersebut. File ini menjelaskan node tindakan dan node alur kontrol alur kerja Anda.

Menghubungkan ke Apache Oozie sebagai sumber

Gunakan prosedur berikut untuk terhubung ke file sumber Apache Oozie Anda.

Untuk terhubung ke Apache Oozie di CLI AWS SCT

1. Buat skrip AWS SCT CLI baru atau edit template skenario yang ada. Misalnya, Anda dapat mengunduh dan mengedit `OozieConversionTemplate.scts` template. Untuk informasi selengkapnya, lihat [Mendapatkan skenario CLI](#).
2. Konfigurasi pengaturan AWS SCT aplikasi.

Contoh kode berikut menyimpan pengaturan aplikasi dan memungkinkan untuk menyimpan password dalam proyek Anda. Anda dapat menggunakan pengaturan tersimpan ini di proyek lain.

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

3. Buat AWS SCT proyek baru.

Contoh kode berikut membuat oozie proyek dalam `c:\sct` folder.

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

4. Tambahkan folder dengan file Apache Oozie sumber Anda ke proyek menggunakan perintah. `AddSource` Pastikan bahwa Anda menggunakan `APACHE_OOZIE` nilai untuk

vendor parameter. Juga, berikan nilai untuk parameter yang diperlukan berikut: name dan mappingsFolder.

Contoh kode berikut menambahkan Apache Oozie sebagai sumber dalam proyek Anda. AWS SCT Contoh ini menciptakan objek sumber dengan nama OZIE. Gunakan nama objek ini untuk menambahkan aturan pemetaan. Setelah Anda menjalankan contoh kode ini, AWS SCT gunakan c:\oozie folder untuk memuat file sumber Anda dalam proyek.

```
AddSource
  -name: 'OOZIE'
  -vendor: 'APACHE_OOZIE'
  -mappingsFolder: 'c:\oozie'
/
```

Anda dapat menggunakan contoh ini dan contoh berikut di Windows.

5. Hubungkan ke file Apache Oozie sumber Anda menggunakan perintah. ConnectSource Gunakan nama objek sumber yang Anda tentukan pada langkah sebelumnya.

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

6. Simpan skrip CLI Anda. Selanjutnya, tambahkan informasi koneksi untuk AWS Step Functions layanan Anda.

Izin untuk menggunakan AWS Lambda fungsi dalam paket ekstensi

Untuk fungsi sumber yang AWS Step Functions tidak mendukung, AWS SCT buat paket ekstensi. Paket ekstensi ini mencakup AWS Lambda fungsi, yang meniru fungsi sumber Anda.

Untuk menggunakan paket ekstensi ini, buat peran AWS Identity and Access Management (IAM) dengan izin berikut.

```
{
  "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:::*/*"
    ]
  }
]
}

```

Untuk menerapkan paket ekstensi, AWS SCT memerlukan peran IAM dengan izin berikut.

```

{
  "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"
    ]
}
]
}

```

Menghubungkan ke AWS Step Functions sebagai target

Gunakan prosedur berikut untuk terhubung AWS Step Functions sebagai target.

Untuk terhubung ke AWS Step Functions di AWS SCT CLI

1. Buka skrip CLI Anda yang mencakup informasi koneksi untuk file sumber Apache Oozie Anda.
2. Tambahkan informasi tentang target migrasi Anda dalam AWS SCT proyek menggunakan `AddTarget` perintah. Pastikan bahwa Anda menggunakan `STEP_FUNCTIONS` nilai untuk `vendor` parameter. Juga, berikan nilai untuk parameter yang diperlukan berikut: `name` dan `profile`.

Contoh kode berikut menambahkan AWS Step Functions sebagai sumber dalam AWS SCT proyek Anda. Contoh ini menciptakan objek target dengan nama `AWS_STEP_FUNCTIONS`. Gunakan nama objek ini ketika Anda membuat aturan pemetaan. Juga, contoh ini menggunakan profil AWS SCT layanan yang Anda buat di langkah prasyarat. Pastikan Anda mengganti *profile_name dengan nama* profil Anda.

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

Jika Anda tidak menggunakan profil AWS layanan, pastikan Anda memberikan nilai untuk parameter yang diperlukan berikut: `accessKey`, `secretKey`, `awsRegion`, dan `s3Path`. Gunakan parameter ini untuk menentukan kunci akses AWS rahasia, kunci AWS rahasia Wilayah AWS, dan jalur ke bucket Amazon S3 Anda.

3. Hubungkan untuk AWS Step Functions menggunakan `ConnectTarget` perintah. Gunakan nama objek target yang Anda tentukan pada langkah sebelumnya.

Contoh kode berikut terhubung ke objek `AWS_STEP_FUNCTIONS` target menggunakan profil AWS layanan Anda. Pastikan Anda mengganti *profile_name dengan nama* profil Anda.

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

4. Simpan skrip CLI Anda. Selanjutnya, tambahkan aturan pemetaan dan perintah migrasi. Untuk informasi selengkapnya, lihat [Mengkonversi Apache Oozie ke AWS Step Functions](#).

Menggunakan Azure SQL Database sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengkonversi skema, objek kode, dan kode aplikasi dari Azure SQL Database ke target berikut:

- Amazon RDS for MySQL
- Amazon Aurora Edisi Kompatibel MySQL
- Amazon RDS for PostgreSQL
- Amazon Aurora Edisi Kompatibel PostgreSQL

Topik

- [Hak istimewa untuk Azure SQL Database sebagai sumber](#)
- [Menghubungkan ke Azure SQL Database sebagai sumber](#)

Hak istimewa untuk Azure SQL Database sebagai sumber

Hak istimewa yang diperlukan untuk Azure SQL Database sebagai sumber adalah sebagai berikut:

- LIHAT DEFINISI
- LIHAT KEADAAN DATABASE

Ulangi hibah untuk setiap database yang skema Anda konversi.

Hak istimewa yang diperlukan untuk database MySQL dan PostgreSQL target dijelaskan di bagian berikut.

- [Hak istimewa untuk MySQL sebagai basis data target](#)
- [Hak istimewa untuk PostgreSQL sebagai basis data target](#)

Menghubungkan ke Azure SQL Database sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Azure SQL Database Anda dengan file. AWS Schema Conversion Tool

Untuk menyambung ke database sumber Azure SQL Database

1. DiAWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Azure SQL Database, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Rahasia, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi database sumber Azure SQL Database secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama server	Masukkan nama Domain Name Service (DNS) atau alamat IP server database sumber Anda.
Basis data	Masukkan nama database untuk terhubung ke.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan</p>

Parameter	Action
	membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.
Simpan kata sandi	AWS SCTmenciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.

- Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
- Pilih Hubungkan untuk terhubung ke database sumber Anda.

Menggunakan IBM Db2 untuk z/OS sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengkonversi skema, objek kode, dan kode aplikasi dari IBM Db2 untuk z/OS ke target berikut.

- Amazon RDS for MySQL
- Amazon Aurora Edisi Kompatibel MySQL
- Amazon RDS for PostgreSQL
- Amazon Aurora Edisi Kompatibel PostgreSQL

Prasyarat untuk Db2 untuk z/OS sebagai database sumber

IBM Db2 untuk z/OS versi 12 fungsi tingkat 100 versi database tidak mendukung sebagian besar kemampuan baru IBM Db2 untuk z/OS versi 12. Versi database ini menyediakan dukungan untuk fallback ke Db2 versi 11 dan berbagi data dengan Db2 versi 11. Untuk menghindari konversi fitur yang tidak didukung dari Db2 versi 11, kami sarankan Anda menggunakan IBM Db2 untuk z/OS fungsi database tingkat 500 atau lebih tinggi sebagai sumber untuk. AWS SCT

Anda dapat menggunakan contoh kode berikut untuk memeriksa versi sumber Anda IBM Db2 untuk database Z/OS.

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

Pastikan bahwa kode ini mengembalikan versi DSN12015 atau lebih tinggi.

Anda dapat menggunakan contoh kode berikut untuk memeriksa nilai register APPLICATION COMPATIBILITY khusus di sumber Anda IBM Db2 untuk database Z/OS.

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

Pastikan bahwa kode ini mengembalikan versi V12R1M500 atau lebih tinggi.

Hak istimewa untuk Db2 untuk z/OS sebagai database sumber

Hak istimewa yang diperlukan untuk terhubung ke Db2 untuk database Z/OS dan membaca katalog sistem dan tabel adalah sebagai berikut:

- PILIH DI SYSIBM.LOCATIONS
- PILIH DI SYSIBM.SYSCHECKS
- PILIH PADA SYSIBM.SYSCOLUMNS
- PILIH DI SYSIBM.SYSDATABASE
- PILIH PADA SYSIBM.SYSDATATYPES
- PILIH PADA SYSIBM.SYSDUMMY1
- PILIH PADA SYSIBM.SYSFOREIGNKEYS
- PILIH PADA SYSIBM.SYSINDEXES
- PILIH PADA SYSIBM.SYSKEYCOLUSE
- PILIH PADA SYSIBM.SYSKEYS
- PILIH PADA SYSIBM.SYSKEYTARGETS
- PILIH PADA SYSIBM.SYSJAROBJECTS
- PILIH PADA SYSIBM.SYSPACKAGE
- PILIH DI SYSIBM.SYSPARMS
- PILIH PADA SYSIBM.SYSRELS
- PILIH DI SYSIBM.SYSROUTINES
- PILIH DI SYSIBM.SYSSEQUENCES
- PILIH DI SYSIBM.SYSSEQUENCESDEP

- PILIH PADA SYSIBM.SYSSYNONYMS
- PILIH DI SYSIBM.SYSTABCONST
- PILIH PADA SYSIBM.SYSTABLES
- PILIH DI SYSIBM.SYSTABLESPACE
- PILIH PADA SYSIBM.SYSTRIGGERS
- PILIH PADA SYSIBM.SYSVARIABLES
- PILIH DI SYSIBM.SYSVIEWS

Untuk mengkonversi Db2 untuk tabel z/OS untuk PostgreSQL dipartisi tabel, mengumpulkan statistik pada tablespaces dan tabel dalam database Anda menggunakan utilitas seperti yang ditunjukkan berikut. RUNSTATS

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

Pada contoh sebelumnya, ganti *YOURDB* placeholder dengan nama database sumber.

Menghubungkan ke Db2 untuk z/OS sebagai sumber

Gunakan prosedur berikut untuk terhubung ke Db2 Anda untuk database sumber z/OS dengan. AWS SCT

Untuk terhubung ke IBM Db2 untuk database sumber z/OS

1. DiAWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Db2 untuk z/OS, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCTmenampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:

1. Untuk AWS Rahasia, pilih nama rahasianya.
2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan IBM Db2 untuk z/OS sumber informasi koneksi database secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama server	Masukkan nama Domain Name System (DNS) atau alamat IP server database sumber Anda.
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Lokasi	Masukkan nama unik dari lokasi Db2 yang ingin Anda akses.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>

Parameter	Action
Gunakan SSL	<p>Pilih opsi ini jika Anda ingin menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none"> • Toko kepercayaan: Lokasi toko perwalian yang berisi sertifikat. Agar lokasi ini muncul di sini, pastikan untuk menambahkannya di Pengaturan global.
Simpan kata sandi	<p>AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.</p>
Db2 untuk jalur driver z/OS	<p>Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

5. Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
6. Pilih Hubungkan untuk terhubung ke database sumber Anda.

Hak istimewa untuk MySQL sebagai database target

Hak istimewa yang diperlukan untuk MySQL sebagai target adalah sebagai berikut:

- BUAT PADA * . *
- MENGUBAH PADA * . *
- MENJATUHKAN * . *
- INDEKS PADA * . *

- REFERENSI PADA* . *
- PILIH PADA* . *
- BUAT TAMPILAN DI* . *
- TAMPILKAN TAMPILAN DI* . *
- PEMICU PADA* . *
- BUAT RUTINITAS DI* . *
- MENGUBAH RUTINITAS PADA* . *
- MENGEKSEKUSI PADA* . *
- PILIH DI mysql.proc
- INSERT, UPDATE PADA AWS_DB2ZOS_EXT. *
- MENYISIPKAN, MEMPERBARUI, MENGHAPUS PADA AWS_DB2ZOS_EXT_DATA. *
- MEMBUAT TABEL SEMENTARA PADA AWS_DB2ZOS_EXT_DATA. *

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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';
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti kata *sandi Anda* dengan kata *sandi* yang aman.

Untuk menggunakan Amazon RDS untuk MySQL sebagai target, tetapkan `log_bin_trust_function_creators` parameter ke `true`, dan `character_set_server` latin1 Untuk mengkonfigurasi parameter ini, membuat grup parameter DB baru atau memodifikasi grup parameter DB yang ada.

Untuk menggunakan Aurora MySQL sebagai target, tetapkan `log_bin_trust_function_creators` parameter ke `true`, dan `character_set_server` latin1 Juga, atur `lower_case_table_names` parameter ke `true`. Untuk mengkonfigurasi parameter ini, membuat grup parameter DB baru atau memodifikasi grup parameter DB yang ada.

Hak istimewa untuk PostgreSQL sebagai database target

Untuk menggunakan PostgreSQL sebagai target, AWS SCT memerlukan hak istimewa. `CREATE ON DATABASE` Pastikan Anda memberikan hak istimewa ini untuk setiap database PostgreSQL target.

Untuk menggunakan Amazon RDS untuk PostgreSQL sebagai target, AWS SCT memerlukan hak istimewa. `rds_superuser`

Untuk menggunakan sinonim publik yang dikonversi, ubah jalur pencarian default basis data ke `"$user"`, `public_synonyms`, `public`.

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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;
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *db_name* dengan nama database target Anda. Terakhir, ganti kata *sandi Anda dengan kata sandi* yang aman.

Di PostgreSQL, hanya pemilik skema atau `superuser` dapat menjatuhkan skema. Pemilik dapat menjatuhkan skema dan semua objek yang disertakan skema ini bahkan jika pemilik skema tidak memiliki beberapa objeknya.

Bila Anda menggunakan pengguna yang berbeda untuk mengkonversi dan menerapkan skema yang berbeda ke database target Anda, Anda bisa mendapatkan pesan kesalahan ketika tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan galat ini, gunakan `superuser` peran.

Db2 untuk z/OS ke pengaturan konversi PostgreSQL

Untuk mengedit pengaturan konversi Db2 untuk z/OS ke PostgreSQL, pilih Pengaturan, dan kemudian pilih Pengaturan konversi. Dari daftar atas, pilih Db2 untuk z/OS, lalu pilih Db2 untuk z/OS - PostgreSQL atau Db2 untuk z/OS - Amazon Aurora (kompatibel dengan PostgreSQL). AWS SCT menampilkan semua pengaturan yang tersedia untuk IBM Db2 untuk z/OS untuk konversi PostgreSQL.

Db2 untuk z/OS ke pengaturan konversi PostgreSQL di AWS SCT termasuk pilihan untuk berikut:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menghasilkan nama unik untuk kendala dalam database target.

Di PostgreSQL, semua nama kendala yang Anda gunakan harus unik. AWS SCT dapat menghasilkan nama unik untuk kendala dalam kode yang dikonversi dengan menambahkan awalan dengan nama tabel ke nama kendala Anda. Untuk memastikan bahwa AWS SCT menghasilkan nama unik untuk kendala Anda, pilih Buat nama unik untuk batasan.

- Untuk menjaga format nama kolom, ekspresi, dan klausa dalam pernyataan DML dalam kode dikonversi.

AWS SCT dapat menjaga tata letak nama kolom, ekspresi, dan klausa dalam pernyataan DML dalam posisi dan urutan yang sama seperti dalam kode sumber. Untuk melakukannya, pilih Ya untuk Jauhkan format nama kolom, ekspresi, dan klausa dalam pernyataan DML.

- Untuk mengecualikan partisi tabel dari lingkup konversi.

AWS SCT dapat melewati semua partisi dari tabel sumber selama konversi. Untuk melakukannya, pilih Kecualikan partisi tabel dari cakupan konversi.

- Untuk menggunakan partisi otomatis untuk tabel yang dipartisi oleh pertumbuhan.

Untuk migrasi data, AWS SCT dapat secara otomatis partisi semua tabel yang lebih besar dari ukuran yang ditentukan. Untuk menggunakan opsi ini, pilih Enforce partisi tabel yang lebih besar dari, dan masukkan ukuran tabel dalam gigabyte. Selanjutnya, masukkan jumlah partisi. AWS SCT mempertimbangkan ukuran perangkat penyimpanan akses langsung (DASD) dari database sumber Anda saat Anda mengaktifkan opsi ini.

AWS SCT dapat menentukan jumlah partisi secara otomatis. Untuk melakukannya, pilih Tingkatkan jumlah partisi secara proporsional, dan masukkan jumlah partisi maksimum.

- Untuk mengembalikan hasil dinamis set sebagai array nilai dari tipe data refkursor.

AWS SCT dapat mengubah prosedur sumber yang mengembalikan set hasil dinamis ke dalam prosedur yang memiliki array refkursor terbuka sebagai parameter output tambahan. Untuk melakukannya, pilih Gunakan larik refkursor untuk mengembalikan semua kumpulan hasil dinamis.

- Untuk menentukan standar yang akan digunakan untuk konversi nilai tanggal dan waktu menjadi representasi string.

AWS SCT dapat mengonversi nilai tanggal dan waktu menjadi representasi string menggunakan salah satu format industri yang didukung. Untuk melakukannya, pilih Gunakan representasi string dari nilai tanggal atau Gunakan representasi string dari nilai waktu. Selanjutnya, pilih salah satu dari standar berikut.

- Organisasi Standar Internasional (ISO)
- IBM Standar Eropa (EUR)
- IBM USA Standard (AS)
- Era Kristen Standar Industri Jepang (JIS)

Menggunakan IBM Db2 LUW sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengkonversi skema, objek kode dalam bahasa SQL, dan kode aplikasi dari IBM Db2 untuk Linux, Unix, dan Windows (Db2 LUW) ke target berikut.

- Amazon RDS for MySQL
- Amazon Aurora Edisi Kompatibel MySQL
- Amazon RDS for PostgreSQL
- Amazon Aurora Edisi Kompatibel PostgreSQL
- Amazon RDS for MariaDB

AWS SCT mendukung sebagai sumber Db2 LUW versi 9.1, 9.5, 9.7, 10.1, 10.5, 11.1, dan 11.5.

Hak istimewa untuk Db2 LUW sebagai sumber

Hak istimewa yang diperlukan untuk terhubung ke database Db2 LUW, untuk memeriksa hak istimewa yang tersedia dan membaca metadata skema untuk sumber adalah sebagai berikut:

- Privilege diperlukan untuk membuat koneksi:
 - TERHUBUNG PADA DATABASE
- Privilege diperlukan untuk menjalankan pernyataan SQL:
 - MENGEKSEKUSI PADA PAKET NULLID.SYSSH200
- Hak istimewa yang diperlukan untuk mendapatkan informasi tingkat instans:
 - MENGEKSEKUSI PADA FUNGSI SYSPROC.ENV_GET_INST_INFO
 - PILIH PADA SYSIBMADM.ENV_INST_INFO
 - PILIH DI SYSIBMADM.ENV_SYS_INFO
- Hak istimewa yang diperlukan untuk memeriksa hak istimewa yang diberikan melalui peran, kelompok, dan otoritas:
 - MENGEKSEKUSI PADA FUNGSI SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID
 - MENGEKSEKUSI PADA FUNGSI SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID
 - MENGEKSEKUSI PADA FUNGSI SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID
 - PILIH DI SYSIBMADM.PRIVILEGES
- Hak istimewa yang dibutuhkan pada katalog dan tabel sistem:
 - PILIH PADA SYSCAT.ATTRIBUTES
 - PILIH DI SYSCAT.CHECKS
 - PILIH PADA SYSCAT.COLIDENTATTRIBUTES
 - PILIH PADA SYSCAT.COLUMNS
 - PILIH PADA SYSCAT.DATAPARTITIONEXPRESSION
 - PILIH DI SYSCAT.DATAPARTITIONS
 - PILIH DI SYSCAT.DATATYPEDEP
 - PILIH PADA SYSCAT.DATATYPES
 - PILIH PADA SYSCAT.HIERARCHIES
 - PILIH DI SYSCAT.INDEXCOLUSE
 - PILIH DI SYSCAT.INDEXES

- PILIH DI SYSCAT.INDEXPARTITIONS
- PILIH DI SYSCAT.KEYCOLUSE
- PILIH PADA SYSCAT.MODULEOBJECTS
- PILIH PADA SYSCAT.MODULES
- PILIH DI SYSCAT.NICKS
- PILIH DI SYSCAT.PERIODS
- PILIH DI SYSCAT.REFERENCES
- PILIH DI SYSCAT.ROUTINEPARMS
- PILIH DI SYSCAT.ROUTINES
- PILIH DI SYSCAT.ROWFIELDS
- PILIH PADA SYSCAT.SCHEMATA
- PILIH DI SYSCAT.SEQUENCES
- PILIH DI SYSCAT.TABCONST
- PILIH PADA SYSCAT.TABLES
- PILIH PADA SYSCAT.TRIGGERS
- PILIH PADA SYSCAT.VARIABLEDEP
- PILIH PADA SYSCAT.VARIABLES
- PILIH DI SYSCAT.VIEWS
- PILIH PADA SYSIBM.SYSDUMMY1
- Untuk menjalankan pernyataan SQL, akun pengguna memerlukan hak istimewa untuk menggunakan setidaknya satu beban kerja yang diaktifkan dalam database. Jika tidak ada beban kerja yang ditetapkan ke pengguna, pastikan bahwa beban kerja pengguna default dapat diakses oleh pengguna:
 - PENGGUNAAN PADA BEBAN KERJA SYSDEFAULTUSERWORKLOAD

Untuk menjalankan kueri, Anda perlu membuat tablespace sementara sistem dengan ukuran halaman 8K, 16K, dan 32K, jika tidak ada. Untuk membuat tablespace sementara, jalankan skrip berikut.

```
CREATE BUFFERPOOL BP8K
IMMEDIATE
ALL DBPARTITIONNUMS
SIZE AUTOMATIC
```

```
NUMBLOCKPAGES 0
PAGE_SIZE 8K;

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_8K
PAGE_SIZE 8192
BUFFERPOOL BP8K;

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

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_BP16K
PAGE_SIZE 16384
BUFFERPOOL BP16K;

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

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

Menghubungkan ke Db2 LUW sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Db2 LUW Anda dengan AWS Schema Conversion Tool

Untuk menyambung ke database sumber Db2 LUW

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Db2 LUW, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.

4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:

- Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:

1. Untuk AWSRahasia, pilih nama rahasianya.
2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi database sumber IBM Db2 LUW secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama server	Masukkan nama Domain Name System (DNS) atau alamat IP server database sumber Anda.
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis data	Masukkan nama database Db2 LUW.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>

Parameter	Action
Gunakan SSL	<p>Pilih opsi ini jika Anda ingin menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none"> • Toko kepercayaan: Lokasi toko perwalian yang berisi sertifikat. Agar lokasi ini muncul di sini, pastikan untuk menambahkannya di Pengaturan global.
Simpan kata sandi	<p>AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.</p>
Jalur pengemudi Db2 LUW	<p>Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

5. Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
6. Pilih Hubungkan untuk terhubung ke database sumber Anda.

Mengonversi Db2 LUW ke Amazon RDS untuk Edisi yang kompatibel dengan PostgreSQL atau Amazon Aurora PostgreSQL

Ketika Anda bermigrasi IBM Db2 LUW ke PostgreSQL, AWS SCT dapat mengkonversi berbagai pernyataan pemicu digunakan dengan Db2 LUW. Pernyataan pemicu ini meliputi:

- Peristiwa pemicu - INSERT, DELETE, dan UPDATE memicu peristiwa menentukan bahwa tindakan dipicu berjalan setiap kali acara diterapkan ke tabel subjek atau tampilan subjek.

Anda dapat menentukan kombinasi acara INSERT, DELETE, dan UPDATE, tetapi Anda dapat menentukan setiap acara hanya sekali. AWS SCT mendukung peristiwa pemicu tunggal dan ganda. Untuk acara, PostgreSQL memiliki fungsi yang hampir sama.

- Event OF COLUMN - Anda dapat menentukan nama kolom dari tabel dasar. Pemicu diaktifkan hanya dengan update kolom yang diidentifikasi dalam daftar kolom-nama. PostgreSQL memiliki fungsi yang sama.
- Pernyataan memicu - Ini menentukan bahwa tindakan dipicu diterapkan hanya sekali untuk seluruh pernyataan. Anda tidak dapat menentukan jenis pemicu granularity untuk pemicu SEBELUM atau BUKAN pemicu. Jika ditentukan, pemicu UPDATE atau DELETE diaktifkan, bahkan jika tidak ada baris yang terpengaruh. PostgreSQL juga memiliki fungsi ini dan memicu deklarasi untuk pemicu pernyataan identik untuk PostgreSQL dan Db2 LUW.
- Klausula referensi - Ini menentukan nama korelasi untuk variabel transisi dan nama tabel untuk tabel transisi. Nama korelasi mengidentifikasi baris tertentu dalam kumpulan baris yang dipengaruhi oleh operasi SQL yang memicu. Nama tabel mengidentifikasi kumpulan lengkap baris yang terpengaruh. Setiap baris yang dipengaruhi oleh operasi SQL pemicu tersedia untuk tindakan yang dipicu oleh kolom kualifikasi dengan nama korelasi yang ditentukan. PostgreSQL tidak mendukung fungsi ini, dan hanya menggunakan nama korelasi BARU atau LAMA.
- BUKAN pemicu - AWS SCT mendukung ini.

Mengkonversi Db2 LUW tabel dipartisi ke PostgreSQL versi 10 tabel dipartisi

AWS SCT dapat mengkonversi tabel Db2 LUW ke tabel dipartisi di PostgreSQL 10. Ada beberapa batasan saat mengkonversi tabel dipartisi Db2 LUW ke PostgreSQL:

- Anda dapat membuat tabel dipartisi dengan kolom nullable di Db2 LUW, dan Anda dapat menentukan partisi untuk menyimpan nilai-nilai NULL. Namun, PostgreSQL tidak mendukung nilai NULL untuk partisi RANGE.
- Db2 LUW dapat menggunakan klausa INCLUSIVE atau EKSKLUSIF untuk mengatur nilai batas jangkauan. PostgreSQL hanya mendukung INCLUSIVE untuk batas awal dan EKSKLUSIF untuk batas akhir. Nama partisi yang dikonversi dalam format `<original_table_name>_<original_partition_name>`.
- Anda dapat membuat kunci primer atau unik untuk tabel dipartisi di Db2 LUW. PostgreSQL mengharuskan Anda untuk membuat kunci primer atau unik untuk setiap partisi secara langsung. Kendala kunci primer atau unik harus dihapus dari tabel induk. Nama kunci yang dikonversi adalah dalam format `<original_key_name>_<original_partition_name>`.

- Anda dapat membuat kendala kunci asing dari dan ke tabel dipartisi di Db2 LUW. Namun, PostgreSQL tidak mendukung referensi kunci asing dalam tabel yang dipartisi. PostgreSQL juga tidak mendukung referensi kunci asing dari tabel yang dipartisi ke tabel lain.
- Anda dapat membuat indeks pada tabel dipartisi di Db2 LUW. Namun, PostgreSQL mengharuskan Anda untuk membuat indeks untuk setiap partisi secara langsung. Indeks harus dihapus dari tabel induk. Nama indeks yang dikonversi dalam format `<original_index_name>_<original_partition_name>`.
- Anda harus menentukan pemicu baris pada partisi individu, bukan pada tabel dipartisi. Pemicu harus dihapus dari tabel induk. Nama pemicu yang dikonversi ada dalam format `<original_trigger_name>_<original_partition_name>`.

Hak istimewa untuk PostgreSQL sebagai target

Untuk menggunakan PostgreSQL sebagai target, AWS SCT memerlukan hak istimewa. `CREATE ON DATABASE` Pastikan Anda memberikan hak istimewa ini untuk setiap database PostgreSQL target.

Untuk menggunakan sinonim publik yang dikonversi, ubah jalur pencarian default basis data ke `"$user", public_synonyms, public`.

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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;
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *db_name* dengan nama database target Anda. Terakhir, ganti kata *sandi Anda dengan kata sandi* yang aman.

Di PostgreSQL, hanya pemilik skema atau `superuser` dapat menjatuhkan skema. Pemilik dapat menjatuhkan skema dan semua objek yang disertakan skema ini bahkan jika pemilik skema tidak memiliki beberapa objeknya.

Bila Anda menggunakan pengguna yang berbeda untuk mengkonversi dan menerapkan skema yang berbeda ke database target Anda, Anda bisa mendapatkan pesan kesalahan ketika tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan galat ini, gunakan `superuser` peran.

Mengonversi Db2 LUW ke Amazon RDS untuk MySQL atau Amazon Aurora MySQL

Saat Anda mengonversi database IBM Db2 LUW ke RDS untuk MySQL atau Amazon Aurora MySQL, perhatikan hal berikut.

Hak istimewa untuk MySQL sebagai target

Hak istimewa yang diperlukan untuk MySQL sebagai target adalah sebagai berikut:

- BUAT PADA* . *
- MENGUBAH PADA * . *
- MENJATUHKAN * . *
- INDEKS PADA* . *
- REFERENSI PADA* . *
- PILIH PADA* . *
- BUAT TAMPILAN DI* . *
- TAMPILKAN TAMPILAN DI* . *
- PEMICU PADA* . *
- BUAT RUTINITAS DI* . *
- MENGUBAH RUTINITAS PADA* . *
- MENGEKSEKUSI PADA* . *
- PILIH DI mysql.proc
- INSERT, UPDATE PADA AWS_DB2_EXT. *
- MENYISIPKAN, MEMPERBARUI, MENGHAPUS PADA AWS_DB2_EXT_DATA. *
- MEMBUAT TABEL SEMENTARA PADA AWS_DB2_EXT_DATA. *

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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';
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti kata *sandi Anda* dengan kata sandi yang aman.

Untuk menggunakan Amazon RDS untuk MySQL atau Aurora MySQL sebagai target, tetapkan parameternya. `lower_case_table_names 1` Nilai ini berarti bahwa server MySQL menangani pengidentifikasi nama objek seperti tabel, indeks, pemicu, dan database sebagai kasus sensitif. Jika Anda telah mengaktifkan pencatatan biner di instance target Anda, maka atur `log_bin_trust_function_creators` parameternya¹. Dalam hal ini, Anda tidak perlu menggunakan `DETERMINISTIC`, `READS SQL DATA` atau `NO SQL` karakteristik untuk membuat fungsi yang disimpan. Untuk mengkonfigurasi parameter ini, membuat grup parameter DB baru atau memodifikasi grup parameter DB yang ada.

Menggunakan MySQL sebagai sumber AWS SCT

Anda dapat menggunakan AWS SCT untuk mengkonversi skema, objek kode database, dan kode aplikasi dari MySQL ke target berikut:

- Amazon RDS for PostgreSQL
- Amazon Aurora Edisi Kompatibel PostgreSQL
- Amazon RDS for MySQL

Untuk informasi selengkapnya, lihat bagian berikut:

Topik

- [Hak untuk MySQL sebagai basis data](#)
- [Menghubungkan ke MySQL sebagai sumber](#)
- [Hak untuk PostgreSQL sebagai basis data target](#)

Hak untuk MySQL sebagai basis data

Hak istimewa yang diperlukan untuk MySQL sebagai sumber adalah sebagai berikut:

- PILIH PADA* . *
- TAMPILKAN TAMPILAN DI* . *

Menghubungkan ke MySQL sebagai sumber

Gunakan prosedur berikut untuk terhubung ke basis data sumber MySQL Anda dengan AWS Schema Conversion Tool.

Untuk terhubung ke basis data sumber MySQL

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih MySQL, lalu pilih Berikutnya.

Kotak dialog Tambahkan sumber akan muncul.

3. Untuk nama koneksi, masukkan nama untuk basis data Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Secret, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Secrets Manager.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat [Menggunakan AWS Secrets Manager](#).

- Untuk memasukkan informasi koneksi database sumber MySQL secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama server	<p>Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.</p> <p>Anda dapat terhubung ke basis data MySQL sumber Anda menggunakan protokol alamat IPv6. Untuk melakukannya, pastikan bahwa Anda menggunakan tanda kurung kuadrat untuk memasukkan alamat IP, seperti yang ditunjukkan dalam contoh berikut.</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>
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server basis data sumber Anda.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi basis data untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>

Parameter	Action
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke basis data Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none"> • Memerlukan SSL: Pilih opsi ini untuk terhubung ke server hanya melalui SSL. <p>Jika Anda memilih Memerlukan SSL, itu berarti bahwa jika server tidak mendukung SSL, Anda tidak dapat menyambung ke server. Jika Anda tidak memilih Memerlukan SSL dan server tidak mendukung SSL, Anda masih dapat terhubung ke server tanpa menggunakan SSL. Untuk informasi selengkapnya, lihat Mengkonfigurasi MySQL untuk Menggunakan Koneksi Aman.</p> <ul style="list-style-type: none"> • Verifikasi sertifikat server: Pilih opsi ini untuk memverifikasi sertifikat server menggunakan toko kepercayaan. • Toko kepercayaan: Lokasi toko perwalian yang berisi sertifikat.
Simpan kata sandi	<p>AWS SCT menciptakan brankas aman untuk menyimpan sertifikat SSL dan kata sandi database. Mengaktifkan opsi ini memungkinkan Anda menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.</p>
MySQL jalur pengemudi	<p>Masukkan jalur ke pengemudi yang akan digunakan untuk terhubung ke basis data sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

5. Pilih Uji Koneksi untuk memverifikasi bahwa Anda AWS SCT dapat terhubung ke basis data sumber Anda.

6. Pilih Connect untuk terhubung ke basis data sumber Anda.

Hak untuk PostgreSQL sebagai basis data target

Untuk menggunakan PostgreSQL sebagai target, AWS SCT memerlukan `CREATE ON DATABASE` hak istimewa. Pastikan Anda memberikan hak istimewa ini untuk setiap database PostgreSQL target.

Untuk menggunakan sinonim publik yang dikonversi, ubah jalur pencarian default basis data ke `"$user", public_synonyms, public`.

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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;
```

Di contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *db_name* dengan nama database target Anda. Terakhir, ganti *kata sandi Anda dengan kata* sandi yang aman.

Di PostgreSQL, hanya pemilik skema atau `superuser` dapat menjatuhkan skema. Pemilik dapat menjatuhkan skema dan semua objek yang disertakan skema ini bahkan jika pemilik skema tidak memiliki beberapa objeknya.

Bila Anda menggunakan pengguna yang berbeda untuk mengkonversi dan menerapkan skema yang berbeda ke database target Anda, Anda bisa mendapatkan pesan kesalahan ketika tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan galat ini, gunakan `superuser` peran.

Menggunakan Oracle Database sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengkonversi skema, objek kode database, dan kode aplikasi dari Oracle Database ke target berikut:

- Amazon RDS for MySQL
- Amazon Aurora Edisi Kompatibel MySQL
- Amazon RDS for PostgreSQL
- Amazon Aurora Edisi Kompatibel PostgreSQL

- Amazon RDS for Oracle
- Amazon RDS for MariaDB

Ketika sumber adalah database Oracle, komentar dapat dikonversi ke format yang sesuai di, misalnya, database PostgreSQL. AWS SCT dapat mengkonversi komentar pada tabel, pandangan, dan kolom. Komentar dapat mencakup apostrof; AWS SCT menggandakan tanda kutip saat mengonversi pernyataan SQL, seperti halnya untuk string literal.

Untuk informasi selengkapnya, lihat berikut ini.

Topik

- [Hak istimewa untuk Oracle sebagai sumber](#)
- [Menghubungkan ke Oracle sebagai sumber](#)
- [Mengonversi Oracle ke Amazon RDS untuk PostgreSQL atau Amazon Aurora PostgreSQL](#)
- [Mengonversi Oracle ke Amazon RDS untuk MySQL atau Amazon Aurora MySQL](#)
- [Mengonversi Oracle ke Amazon RDS untuk Oracle](#)

Hak istimewa untuk Oracle sebagai sumber

Hak istimewa yang diperlukan untuk Oracle sebagai sumber adalah sebagai berikut:

- MENGHUBUNG
- SELECT_CATALOG_ROLE
- PILIH KAMUS APA SAJA
- PILIH DI SYS.ARGUMENT\$

Menghubungkan ke Oracle sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Oracle Anda dengan AWS Schema Conversion Tool

Untuk menyambung ke database sumber Oracle

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Oracle, lalu pilih Berikutnya.

Kotak dialog Add source muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensial database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Secret, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi database sumber Oracle secara manual, gunakan petunjuk berikut:

Parameter	Action
Jenis	<p>Pilih jenis koneksi ke database Anda. Tergantung pada jenis Anda, berikan informasi tambahan berikut:</p> <ul style="list-style-type: none"> • SID <ul style="list-style-type: none"> • Nama server: Nama Domain Name System (DNS) atau alamat IP server database sumber Anda. • Port server: Port yang digunakan untuk terhubung ke server database sumber Anda. • Oracle SID: Oracle Sistem ID (SID). Untuk menemukan Oracle SID, kirimkan query berikut ke database Oracle Anda: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre> • Nama layanan <ul style="list-style-type: none"> • Nama server: Nama DNS atau alamat IP server database sumber Anda.

Parameter	Action
	<p>Anda dapat terhubung ke database Oracle sumber Anda menggunakan protokol alamat IPv6. Untuk melakukannya, pastikan Anda menggunakan tanda kurung persegi untuk memasukkan alamat IP, seperti yang ditunjukkan pada contoh berikut.</p> <div data-bbox="721 474 1507 554" style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; text-align: center;"><code>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</code></div> <ul style="list-style-type: none">• Port server: Port yang digunakan untuk terhubung ke server database sumber Anda.• Nama layanan: Nama layanan Oracle untuk terhubung ke.• Alias TNS<ul style="list-style-type: none">• Jalur file TNS: Jalur ke file yang berisi informasi koneksi nama Transparan Jaringan Substrat (TNS). <p>Setelah Anda memilih file TNS, AWS SCT tambahkan semua koneksi database Oracle dari file ke daftar alias TNS.</p> <p>Pilih opsi ini untuk terhubung ke Oracle Real Application Cluster (RAC).</p> <ul style="list-style-type: none">• Alias TNS: Alias TNS dari file ini untuk digunakan untuk terhubung ke database sumber.• TNS menghubungkan identifier<ul style="list-style-type: none">• TNS connect identifier: Pengenal untuk informasi koneksi TNS terdaftar.

Parameter	Action
Nama pengguna dan Kata Sandi	<p>Masukkan kredensial database untuk terhubung ke server database sumber Anda.</p> <p>Pertama kali Anda terhubung ke database Oracle, Anda memasukkan path ke file Oracle Driver (ojdbc8.jar). Anda dapat mengunduh file di http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html. Pastikan untuk mendaftar di situs web Oracle Technical Network gratis untuk menyelesaikan unduhan. AWS SCT menggunakan driver yang dipilih untuk setiap koneksi database Oracle masa depan. Jalur driver dapat dimodifikasi menggunakan tab Driver di Pengaturan Global.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none">• Otentikasi SSL: Pilih opsi ini untuk menggunakan otentikasi SSL berdasarkan sertifikat Siapkan toko kepercayaan dan penyimpanan kunci Anda di Pengaturan global, Keamanan.• Toko kepercayaan: Toko kepercayaan untuk digunakan.• Toko kunci: Toko kunci untuk digunakan.

Parameter	Action
Simpan kata sandi	AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Pilih opsi ini untuk menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.
Jalur pengemudi Oracle	Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan . Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global .

- Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
- Pilih Hubungkan untuk terhubung ke database sumber Anda.

Mengonversi Oracle ke Amazon RDS untuk PostgreSQL atau Amazon Aurora PostgreSQL

Saat Anda mengonversi database Oracle ke RDS untuk PostgreSQL atau Amazon Aurora PostgreSQL, perhatikan hal berikut.

Topik

- [Hak istimewa untuk PostgreSQL sebagai database target](#)
- [Pengaturan konversi Oracle ke PostgreSQL](#)
- [Mengkonversi urutan Oracle](#)
- [Mengkonversi Oracle ROWID](#)
- [Mengkonversi Oracle SQL dinamis](#)
- [Mengkonversi partisi Oracle](#)

Ketika mengkonversi objek sistem Oracle ke PostgreSQL, AWS SCT melakukan konversi seperti yang ditunjukkan pada tabel berikut.

Objek sistem Oracle	Deskripsi	Dikonversi objek PostgreSQL
V \$ VERSI	Menampilkan nomor versi komponen pustaka inti di Oracle Database	aws_oracle_ext.v \$ versi
V \$ CONTOH	Tampilan yang menunjukkan keadaan instance saat ini.	aws_oracle_ext.v \$ contoh

Anda dapat menggunakan AWS SCT untuk mengkonversi file Oracle SQL* Plus ke psql, yang merupakan front-end berbasis terminal untuk PostgreSQL. Untuk informasi selengkapnya, lihat [Mengonversi aplikasi SQL menggunakan AWS SCT](#).

Hak istimewa untuk PostgreSQL sebagai database target

Untuk menggunakan PostgreSQL sebagai target, AWS SCT memerlukan hak istimewa. CREATE ON DATABASE Pastikan Anda memberikan hak istimewa ini untuk setiap database PostgreSQL target.

Untuk menggunakan sinonim publik yang dikonversi, ubah jalur pencarian default basis data ke"\$user", public_synonyms, public.

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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;
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *db_name* dengan nama database target Anda. Terakhir, ganti kata *sandi Anda dengan kata sandi* yang aman.

Untuk menggunakan Amazon RDS untuk PostgreSQL sebagai target, AWS SCT memerlukan hak istimewa. rds_superuser

Di PostgreSQL, hanya pemilik skema atau `superuser` dapat menjatuhkan skema. Pemilik dapat menjatuhkan skema dan semua objek yang disertakan skema ini bahkan jika pemilik skema tidak memiliki beberapa objeknya.

Bila Anda menggunakan pengguna yang berbeda untuk mengkonversi dan menerapkan skema yang berbeda ke database target Anda, Anda bisa mendapatkan pesan kesalahan ketika tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan galat ini, gunakan `superuser` peran.

Pengaturan konversi Oracle ke PostgreSQL

Untuk mengedit pengaturan konversi Oracle ke PostgreSQL, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Oracle, lalu pilih Oracle - PostgreSQL. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Oracle ke PostgreSQL.

Pengaturan konversi Oracle ke PostgreSQL di AWS SCT termasuk pilihan untuk berikut:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk memungkinkan AWS SCT untuk mengkonversi Oracle terwujud pandangan ke tabel atau tampilan terwujud pada PostgreSQL. Untuk konversi tampilan terwujud sebagai, pilih cara mengonversi tampilan terwujud sumber Anda.
- Untuk bekerja dengan kode Oracle sumber Anda saat menyertakan `TO_CHAR`, `TO_DATE`, dan `TO_NUMBER` fungsi dengan parameter yang tidak didukung PostgreSQL. Secara default, AWS SCT mengemulasi penggunaan parameter ini dalam kode yang dikonversi.

Ketika kode Oracle sumber Anda hanya menyertakan parameter yang didukung PostgreSQL, Anda dapat menggunakan PostgreSQL `TO_CHAR`, dan fungsi asli. `TO_DATE` `TO_NUMBER` Dalam hal ini, kode yang dikonversi bekerja lebih cepat. Untuk menyertakan hanya parameter ini, pilih nilai berikut:

- Fungsi `TO_CHAR` () tidak menggunakan string format khusus Oracle
- Fungsi `TO_DATE` () tidak menggunakan string format khusus Oracle

- Fungsi `TO_NUMBER ()` tidak menggunakan string format khusus Oracle
- Untuk mengatasi ketika sumber database Oracle menyimpan hanya nilai integer di kolom kunci primer atau asing dari tipe `NUMBER` data, AWS SCT dapat mengkonversi kolom ini ke tipe `BIGINT` data. Pendekatan ini meningkatkan kinerja kode Anda yang dikonversi. Untuk mengambil pendekatan ini, pilih `Convert NUMBER primary/foreign key column ke BIGINT`. Pastikan bahwa sumber Anda tidak menyertakan nilai floating point di kolom ini untuk menghindari kehilangan data.
- Untuk melewati pemicu dan kendala yang dinonaktifkan dalam kode sumber Anda. Untuk melakukannya, pilih `Abaikan pemicu dan batasan yang dinonaktifkan`.
- Untuk menggunakan AWS SCT untuk mengkonversi variabel string yang disebut sebagai SQL dinamis. Kode database Anda dapat mengubah nilai variabel string ini. Untuk memastikan bahwa AWS SCT selalu mengubah nilai terbaru dari variabel string ini, pilih `Mengkonversi kode SQL dinamis yang dibuat dalam disebut rutinitas`.
- Untuk mengatasi PostgreSQL versi 10 dan sebelumnya tidak mendukung prosedur. Jika Anda atau pengguna Anda tidak terbiasa menggunakan prosedur di PostgreSQL, AWS SCT dapat mengonversi prosedur Oracle ke fungsi PostgreSQL. Untuk melakukannya, pilih `Convert procedures to functions`.
- Untuk melihat informasi tambahan tentang item tindakan yang terjadi. Untuk melakukannya, Anda dapat menambahkan fungsi tertentu ke paket ekstensi dengan memilih `Add on exception raise block` untuk masalah migrasi dengan tingkat keparahan berikutnya. Kemudian pilih tingkat keparahan untuk meningkatkan pengecualian yang ditentukan pengguna.
- Untuk bekerja dengan database Oracle sumber yang mungkin menyertakan kendala dengan nama yang dihasilkan secara otomatis. Jika kode sumber menggunakan nama-nama ini, pastikan Anda memilih `Konversikan nama pembatas yang dihasilkan sistem menggunakan nama asli sumber`. Jika kode sumber Anda menggunakan batasan ini tetapi tidak menggunakan namanya, kosongkan opsi ini untuk meningkatkan kecepatan konversi.
- Untuk mengatasi apakah database dan aplikasi Anda berjalan di zona waktu yang berbeda. Secara default, AWS SCT mengemulasi zona waktu dalam kode yang dikonversi. Namun, Anda tidak memerlukan emulasi ini ketika database dan aplikasi Anda menggunakan zona waktu yang sama. Dalam hal ini, pilih `Zona waktu di sisi klien cocok dengan zona waktu di server`.
- Untuk mengatasi apakah database sumber dan target Anda berjalan di zona waktu yang berbeda. Jika ya, fungsi yang mengemulasi fungsi Oracle `SYSDATE` bawaan mengembalikan nilai yang berbeda dibandingkan dengan fungsi sumber. Untuk memastikan bahwa sumber dan fungsi target Anda mengembalikan nilai yang sama, pilih `Tetapkan zona waktu default untuk emulasi SYSDATE`.

- Untuk menggunakan fungsi dari ekstensi orafce dalam kode Anda dikonversi. Untuk melakukannya, untuk implementasi Gunakan orafce, pilih fungsi yang akan digunakan. Untuk informasi lebih lanjut tentang orafce, lihat [orafce](#) on. GitHub

Mengkonversi urutan Oracle

AWS SCT mengkonversi urutan dari Oracle ke PostgreSQL. Jika Anda menggunakan urutan untuk mempertahankan batasan integritas, pastikan bahwa nilai baru dari urutan yang dimigrasi tidak tumpang tindih dengan nilai yang ada.

Untuk mengisi urutan dikonversi dengan nilai terakhir dari database sumber

1. Buka AWS SCT proyek Anda dengan Oracle sebagai sumbernya.
2. Pilih Pengaturan, lalu pilih Pengaturan konversi.
3. Dari daftar atas, pilih Oracle, lalu pilih Oracle - PostgreSQL. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Oracle ke PostgreSQL.
4. Pilih Isi urutan yang dikonversi dengan nilai terakhir yang dihasilkan di sisi sumber.
5. Memilih OK untuk menyimpan pengaturan dan menutup Pengaturan konversi kotak dialog.

Mengkonversi Oracle ROWID

Dalam database Oracle, pseudocolumn ROWID berisi alamat baris tabel. Pseudocolumn ROWID unik untuk Oracle, sehingga AWS SCT mengubah pseudocolumn ROWID ke kolom data di PostgreSQL. Dengan menggunakan konversi ini, Anda dapat menyimpan informasi ROWID.

Ketika mengkonversi pseudocolumn ROWID, AWS SCT dapat membuat kolom data dengan tipe data. `bigint` Jika tidak ada kunci utama ada, AWS SCT menetapkan kolom ROWID sebagai kunci utama. Jika kunci utama ada, AWS SCT atur kolom ROWID dengan kendala unik.

Jika kode database sumber Anda mencakup operasi dengan ROWID, yang tidak dapat Anda jalankan menggunakan tipe data numerik, AWS SCT dapat membuat kolom data dengan tipe `character varying` data.

Untuk membuat kolom data untuk Oracle ROWID untuk sebuah proyek

1. Buka AWS SCT proyek Anda dengan Oracle sebagai sumbernya.
2. Pilih Pengaturan, lalu pilih Pengaturan konversi.

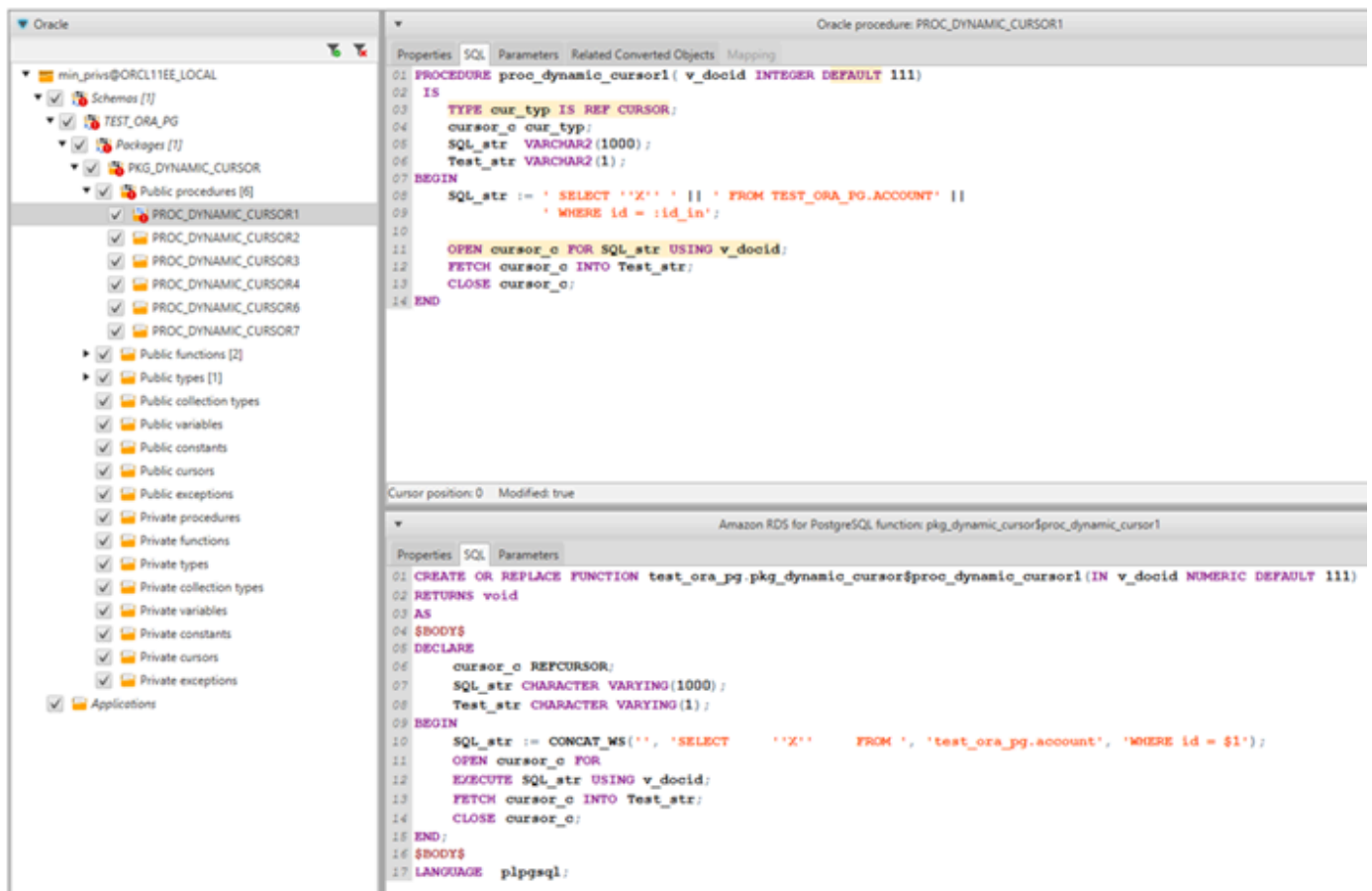
3. Dari daftar atas, pilih Oracle, lalu pilih Oracle - PostgreSQL. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Oracle ke PostgreSQL.
4. Untuk Buat ID baris, lakukan salah satu dari berikut ini:
 - Pilih Hasilkan sebagai identitas untuk membuat kolom data numerik.
 - Pilih Hasilkan sebagai tipe domain karakter untuk membuat kolom data karakter.
5. Memilih OK untuk menyimpan pengaturan dan menutup Pengaturan konversi kotak dialog.

Mengkonversi Oracle SQL dinamis

Oracle menyediakan dua cara untuk menerapkan SQL dinamis: menggunakan pernyataan EXECUTE IMMEDIATE atau memanggil prosedur dalam paket DBMS_SQL. Jika database Oracle sumber Anda menyertakan objek dengan SQL dinamis, gunakan AWS SCT untuk mengonversi pernyataan SQL dinamis Oracle ke PostgreSQL.

Untuk mengkonversi Oracle dynamic SQL ke PostgreSQL

1. Buka AWS SCT proyek Anda dengan Oracle sebagai sumbernya.
2. Pilih objek database yang menggunakan SQL dinamis dalam tampilan pohon sumber Oracle.
3. Buka menu konteks (klik kanan) untuk objek, pilih Convert schema, dan setuju untuk mengganti objek jika ada. Screenshot berikut menunjukkan prosedur yang dikonversi di bawah prosedur Oracle dengan SQL dinamis.



Mengkonversi partisi Oracle

AWS SCT saat ini mendukung metode partisi berikut:

- Kisaran
- Daftar
- Rentang multicolumn
- Hash
- Komposit (daftar-daftar, range-list, daftar-range, daftar-hash, kisaran-hash, hash-hash)

Mengonversi Oracle ke Amazon RDS untuk MySQL atau Amazon Aurora MySQL

Untuk meniru fungsi database Oracle dalam kode MySQL yang telah dikonversi, gunakan paket ekstensi Oracle ke MySQL. AWS SCT Untuk informasi selengkapnya tentang paket ekstensi, lihat [Menggunakan paket AWS SCT ekstensi](#).

Topik

- [Hak istimewa untuk MySQL sebagai database target](#)
- [Pengaturan konversi Oracle ke MySQL](#)
- [Pertimbangan migrasi](#)
- [Mengkonversi pernyataan WITH di Oracle ke RDS untuk MySQL atau Amazon Aurora MySQL](#)

Hak istimewa untuk MySQL sebagai database target

Hak istimewa yang diperlukan untuk MySQL sebagai target adalah sebagai berikut:

- BUAT DI * . *
- MENGUBAH PADA * . *
- JATUHKAN * . *
- INDEKS PADA * . *
- REFERENSI PADA * . *
- PILIH DI * . *
- BUAT TAMPILAN DI* . *
- TAMPILKAN TAMPILAN DI* . *
- MEMICU PADA * . *
- BUAT RUTINITAS DI* . *
- MENGUBAH RUTIN DI * . *
- MENGEKSEKUSI PADA * . *
- MEMBUAT TABEL SEMENTARA DI * . *
- AWS_LAMBDA_ACCESS
- INSERT, UPDATE PADA AWS_ORACLE_EXT. *

- MENYISIPKAN, MEMPERBARUI, MENGHAPUS PADA AWS_ORACLE_EXT_DATA. *

Jika Anda menggunakan database MySQL versi 5.7 atau lebih rendah sebagai target, maka berikan izin INVOKE LAMBDA *.* alih-alih AWS_LAMBDA_ACCESS. Untuk database MySQL versi 8.0 dan yang lebih tinggi, berikan izin AWS_LAMBDA_ACCESS.

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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';
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti kata *sandi* Anda dengan kata sandi yang aman.

Jika Anda menggunakan database MySQL versi 5.7 atau lebih rendah sebagai target, maka gunakan sebagai GRANT INVOKE LAMBDA ON *.* TO '*user_name*' gantinya. GRANT AWS_LAMBDA_ACCESS TO '*user_name*'

Untuk menggunakan Amazon RDS untuk MySQL atau Aurora MySQL sebagai target, tetapkan parameternya. `lower_case_table_names 1` Nilai ini berarti bahwa server MySQL menangani pengidentifikasi nama objek seperti tabel, indeks, pemicu, dan database sebagai kasus sensitif. Jika Anda telah mengaktifkan pencatatan biner di instance target Anda, maka atur `log_bin_trust_function_creators` parameternya¹. Dalam hal ini, Anda tidak perlu menggunakan DETERMINISTIC, READS SQL DATA atau NO SQL karakteristik untuk membuat fungsi

yang tersimpan. Untuk mengonfigurasi parameter ini, buat grup parameter DB baru atau ubah grup parameter DB yang ada.

Pengaturan konversi Oracle ke MySQL

Untuk mengedit pengaturan konversi Oracle ke MySQL, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Oracle, dan kemudian pilih Oracle - MySQL. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Oracle ke MySQL.

Oracle ke pengaturan konversi MySQL di AWS SCT termasuk pilihan untuk berikut:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk mengatasi bahwa database Oracle sumber Anda dapat menggunakan ROWID pseudocolumn tetapi MySQL tidak mendukung fungsi serupa. AWS SCT dapat meniru ROWID pseudocolumn dalam kode dikonversi. Untuk melakukannya, pilih Generate as identity untuk Generate row ID? .

Jika kode Oracle sumber Anda tidak menggunakan **ROWID** pseudocolumn, pilih Jangan buat untuk Buat ID baris? Dalam hal ini, kode yang dikonversi bekerja lebih cepat.

- Untuk bekerja dengan kode Oracle sumber Anda saat menyertakan TO_CHAR, TO_DATE, dan TO_NUMBER fungsi dengan parameter yang tidak didukung MySQL. Secara default, AWS SCT mengemulasi penggunaan parameter ini dalam kode yang dikonversi.

Ketika kode Oracle sumber Anda hanya menyertakan parameter yang didukung PostgreSQL, Anda dapat menggunakan MySQL, TO_CHAR, TO_DATE dan fungsi asli. TO_NUMBER Dalam hal ini, kode yang dikonversi bekerja lebih cepat. Untuk menyertakan hanya parameter ini, pilih nilai berikut:

- Fungsi TO_CHAR () tidak menggunakan string format khusus Oracle
- Fungsi TO_DATE () tidak menggunakan string format khusus Oracle
- Fungsi TO_NUMBER () tidak menggunakan string format khusus Oracle

- Untuk menambahkan apakah database dan aplikasi Anda berjalan di zona waktu yang berbeda. Secara default, AWS SCT mengemulasi zona waktu dalam kode yang dikonversi. Namun, Anda tidak memerlukan emulasi ini ketika database dan aplikasi Anda menggunakan zona waktu yang sama. Dalam hal ini, pilih Zona waktu di sisi klien cocok dengan zona waktu di server.

Pertimbangan migrasi

Saat Anda mengonversi Oracle ke RDS untuk MySQL atau Aurora MySQL, untuk mengubah urutan pernyataan berjalan, Anda dapat menggunakan pernyataan dan label. GOTO Setiap pernyataan PL/SQL yang mengikuti GOTO pernyataan dilewati, dan pemrosesan berlanjut pada label. Anda dapat menggunakan GOTO pernyataan dan label di mana saja dalam prosedur, batch, atau blok pernyataan. Anda juga dapat pernyataan GOTO berikutnya.

MySQL tidak menggunakan GOTO pernyataan. Ketika AWS SCT mengkonversi kode yang berisi GOTO pernyataan, itu mengubah pernyataan untuk menggunakan BEGIN...END atau LOOP...END LOOP pernyataan.

Anda dapat menemukan contoh bagaimana AWS SCT mengkonversi GOTO pernyataan dalam tabel berikut.

Pernyataan Oracle	Pernyataan MySQL
<pre> BEGIN statement1; GOTO label1; statement2; label1: Statement3; END </pre>	<pre> BEGIN label1: BEGIN statement1; LEAVE label1; statement2; END; Statement3; END </pre>
<pre> BEGIN </pre>	<pre> BEGIN </pre>

Pernyataan Oracle	Pernyataan MySQL
<pre>statement1; label1: statement2; GOTO label1; statement3; statement4; END</pre>	<pre>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>

Mengkonversi pernyataan WITH di Oracle ke RDS untuk MySQL atau Amazon Aurora MySQL

Anda menggunakan klausa WITH (subquery_factoring) di Oracle untuk menetapkan nama (query_name) ke blok subquery. Anda kemudian dapat referensi subquery blok beberapa tempat dalam query dengan menentukan query_name. Jika blok subquery tidak berisi link atau parameter

(lokal, prosedur, fungsi, paket), kemudian AWS SCT mengkonversi klausa ke tampilan atau tabel sementara.

Keuntungan dari mengkonversi klausa ke tabel sementara adalah bahwa referensi berulang ke subquery mungkin lebih efisien. Efisiensi yang lebih besar adalah karena data mudah diambil dari tabel sementara daripada yang diperlukan oleh setiap referensi. Anda dapat meniru ini dengan menggunakan tampilan tambahan atau tabel sementara. Nama tampilan menggunakan format <procedure_name>\$.<subselect_alias>.

Anda dapat menemukan contoh dalam tabel berikut.

Pernyataan Oracle	Pernyataan MySQL
<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 (WITH 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 = p_state AND dept_id = l_dept_id) SELECT id,state FROM dept_emp1 ORDER BY id) LOOP NULL; END LOOP;</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); 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_emp1 ORDER BY ID; DECLARE CONTINUE HANDLER FOR NOT FOUND SET done := TRUE; OPEN cur; read_label: LOOP FETCH cur INTO var\$id, var \$state;</pre>

Pernyataan Oracle	Pernyataan MySQL
	<pre>IF done THEN LEAVE read_label; END IF; BEGIN END; END LOOP; CLOSE cur; END;</pre>

Pernyataan Oracle

```

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;

```

Pernyataan MySQL

```

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

```

Pernyataan Oracle	Pernyataan MySQL
	<pre>WHERE dept_emp1.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>

Pernyataan Oracle

```

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;

```

Pernyataan MySQL

```

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:

```

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

Mengonversi Oracle ke Amazon RDS untuk Oracle

Beberapa hal yang perlu dipertimbangkan saat memigrasi skema dan kode Oracle ke Amazon RDS untuk Oracle:

- AWS SCT dapat menambahkan objek direktori ke pohon objek. Objek direktori adalah struktur logis yang masing-masing mewakili direktori fisik pada sistem file server. Anda dapat menggunakan objek direktori dengan paket seperti DBMS_LOB, UTL_FILE, DBMS_FILE_TRANSFER, utilitas DATAPUMP, dan sebagainya.
- AWS SCT mendukung konversi tablespace Oracle ke Amazon RDS untuk instans Oracle DB. Oracle menyimpan data secara logis di tablespace dan secara fisik dalam file data yang terkait dengan tablespace yang sesuai. Di Oracle, Anda dapat membuat tablespace dengan nama file data. Amazon RDS mendukung Oracle Managed Files (OMF) untuk file data, file log, dan file kontrol saja. AWS SCT menciptakan file data yang dibutuhkan selama konversi.
- AWS SCT dapat mengonversi peran dan hak istimewa tingkat server. Mesin database Oracle menggunakan keamanan berbasis peran. Peran adalah kumpulan hak istimewa yang dapat Anda berikan atau dicabut dari pengguna. Peran yang telah ditentukan sebelumnya dalam Amazon RDS, yang disebut DBA, biasanya memungkinkan semua hak administratif pada mesin database Oracle. Hak istimewa berikut ini tidak tersedia untuk peran DBA di instans DB Amazon RDS yang menggunakan mesin Oracle:
 - Ubah basis data
 - Mengubah sistem

- Buat direktori apa pun
- Berikan hak istimewa apa pun
- Berikan peran apa pun
- Buat pekerjaan eksternal

Anda dapat memberikan semua hak istimewa lainnya ke peran pengguna Amazon RDS for Oracle, termasuk hak pemfilteran dan kolom lanjutan.

- AWS SCT mendukung konversi pekerjaan Oracle menjadi pekerjaan yang dapat berjalan di Amazon RDS untuk Oracle. Ada beberapa batasan untuk konversi, termasuk yang berikut ini:
 - Pekerjaan yang dapat dieksekusi tidak didukung.
 - Jadwalkan pekerjaan yang menggunakan tipe data ANYDATA sebagai argumen tidak didukung.
- Oracle Real Application Cluster (RAC) Satu Node adalah pilihan untuk Oracle Database Enterprise Edition yang diperkenalkan dengan Oracle Database 11g Release 2. Amazon RDS untuk Oracle tidak mendukung fitur RAC. Untuk ketersediaan tinggi, gunakan Amazon RDS Multi-AZ.

Dalam penerapan Multi-AZ, Amazon RDS secara otomatis menyediakan dan mempertahankan replika siaga yang sinkron di Availability Zone yang berbeda. Instance DB utama direplikasi secara sinkron di Availability Zones ke replika siaga. Fungsionalitas ini menyediakan redundansi data, menghilangkan pembekuan I/O, dan meminimalkan lonjakan latensi selama pencadangan sistem.

- Oracle Spatial menyediakan skema dan fungsi SQL yang memfasilitasi penyimpanan, pengambilan, pembaruan, dan kueri kumpulan data spasial dalam database Oracle. Oracle Locator menyediakan kemampuan yang biasanya dibutuhkan untuk mendukung aplikasi berbasis internet dan layanan nirkabel serta solusi GIS berbasis mitra. Oracle Locator adalah bagian terbatas dari Oracle Spatial.

Untuk menggunakan fitur Oracle Spatial dan Oracle Locator, tambahkan opsi SPASIAL atau opsi LOCATOR (saling eksklusif) ke grup opsi instans DB Anda.

Ada beberapa prasyarat untuk menggunakan Oracle Spatial dan Oracle Locator pada instans Amazon RDS for Oracle DB:

- Instance harus menggunakan Oracle Enterprise Edition versi 12.1.0.2.v6 atau lebih tinggi, atau 11.2.0.4.v10 atau lebih tinggi.
- Instance harus berada di dalam virtual private cloud (VPC).

- Instance harus kelas instans DB yang dapat mendukung fitur Oracle. Misalnya, Oracle Spatial tidak didukung untuk db.m1.small, db.t1.micro, db.t2.micro, atau db.t2.small kelas instans DB. Untuk informasi selengkapnya, lihat [dukungan kelas instans DB untuk Oracle](#).
- Instance harus mengaktifkan opsi Auto Minor Version Upgrade. Amazon RDS memperbarui instans DB Anda ke Oracle PSU terbaru jika ada kerentanan keamanan dengan skor CVSS 9+ atau kerentanan keamanan lainnya yang diumumkan. Untuk informasi selengkapnya, lihat [Pengaturan untuk instans Oracle DB](#).
- Jika instans DB Anda adalah versi 11.2.0.4.v10 atau lebih tinggi, Anda harus menginstal opsi XMLDB. Untuk informasi selengkapnya, lihat [Oracle XML-DB](#).
- Anda harus memiliki lisensi Oracle Spatial dari Oracle. Untuk informasi lebih lanjut, lihat [Oracle Spatial and Graph](#) dalam dokumentasi Oracle.
- Data Guard disertakan dengan Oracle Database Enterprise Edition. Untuk ketersediaan tinggi, gunakan fitur Multi-AZ Amazon RDS.

Dalam penerapan Multi-AZ, Amazon RDS secara otomatis menyediakan dan mempertahankan replika siaga yang sinkron di Availability Zone yang berbeda. Instance DB utama direplikasi secara sinkron di Availability Zones ke replika siaga. Fungsionalitas ini menyediakan redundansi data, menghilangkan pembekuan I/O, dan meminimalkan lonjakan latensi selama pencadangan sistem.

- AWS SCT mendukung konversi objek Oracle DBMS_SCHEDULER saat bermigrasi ke Amazon RDS untuk Oracle. Laporan AWS SCT penilaian menunjukkan apakah objek jadwal dapat dikonversi. Untuk informasi selengkapnya tentang penggunaan objek jadwal dengan Amazon RDS, lihat dokumentasi [Amazon RDS](#).
- Untuk konversi Oracle ke Amazon RDS untuk Oracle, DB Links didukung. Sebuah link database adalah objek skema dalam satu database yang memungkinkan Anda untuk mengakses objek pada database lain. Database lainnya tidak perlu menjadi database Oracle. Namun, untuk mengakses database non-Oracle Anda harus menggunakan Oracle Heterogeneous Services.

Setelah Anda membuat link database, Anda dapat menggunakan link dalam pernyataan SQL untuk merujuk ke tabel, pandangan, dan PL/SQL objek dalam database lain. Untuk menggunakan link database, tambahkan `@dblink` ke tabel, tampilan, atau PL/SQL nama objek. Anda dapat query tabel atau tampilan dalam database lain dengan pernyataan SELECT. Untuk informasi lebih lanjut tentang menggunakan link database Oracle, lihat dokumentasi [Oracle](#).

Untuk informasi selengkapnya tentang penggunaan tautan database dengan Amazon RDS, lihat dokumentasi [Amazon RDS](#).

- Laporan AWS SCT penilaian menyediakan metrik server untuk konversi. Metrik ini tentang instans Oracle Anda termasuk yang berikut:
 - Perhitungan dan kapasitas memori dari contoh DB target.
 - Fitur Oracle yang tidak didukung seperti Kluster Aplikasi Nyata yang tidak didukung Amazon RDS.
 - Beban baca-tulis disk
 - Total throughput disk rata-rata
 - Informasi server seperti nama server, OS, nama host, dan set karakter.

Hak istimewa untuk RDS untuk Oracle sebagai target

Untuk bermigrasi ke Amazon RDS for Oracle, buat pengguna database dengan hak istimewa. Anda dapat menggunakan contoh kode berikut.

```
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;
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti kata *sandi* Anda dengan kata sandi yang aman.

Batasan saat mengonversi Oracle ke Amazon RDS untuk Oracle

Beberapa batasan yang harus Anda pertimbangkan saat memigrasi skema dan kode Oracle ke Amazon RDS untuk Oracle:

- Peran yang telah ditentukan sebelumnya dalam Amazon RDS, yang disebut DBA, biasanya memungkinkan semua hak administratif pada mesin database Oracle. Hak istimewa berikut ini tidak tersedia untuk peran DBA di instans DB Amazon RDS yang menggunakan mesin Oracle:
 - Ubah basis data
 - Mengubah sistem
 - Buat direktori apa pun
 - Berikan hak istimewa apa pun
 - Berikan peran apa pun
 - Buat pekerjaan eksternal

Anda dapat memberikan semua hak istimewa lainnya ke peran pengguna Oracle RDS.

- Amazon RDS for Oracle mendukung audit tradisional, audit berbutir halus menggunakan paket DBMS_FGA, dan Oracle Unified Auditing.
- Amazon RDS untuk Oracle tidak mendukung pengambilan data perubahan (CDC). Untuk melakukan CDC selama dan setelah migrasi database, gunakan AWS Database Migration Service.

Menggunakan PostgreSQL sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengkonversi skema, objek kode database, dan kode aplikasi dari PostgreSQL ke target berikut:

- Amazon RDS for MySQL
- Amazon Aurora Edisi Kompatibel MySQL
- Amazon RDS for PostgreSQL
- Amazon Aurora Edisi Kompatibel PostgreSQL

Untuk informasi selengkapnya, lihat bagian berikut:

Topik

- [Hak istimewa untuk PostgreSQL sebagai basis data sumber](#)
- [Menghubungkan ke PostgreSQL sebagai sumber](#)
- [Hak istimewa untuk MySQL sebagai basis data target](#)

Hak istimewa untuk PostgreSQL sebagai basis data sumber

Hak istimewa yang diperlukan untuk PostgreSQL sebagai sumber adalah sebagai berikut:

- TERHUBUNG DI DATABASE <database_name>
- PENGGUNAAN PADA SKEMA <database_name>
- PILIH PADA SEMUA TABEL DALAM SKEMA <database_name>
- PILIH PADA SEMUA URUTAN DALAM SKEMA <database_name>

Menghubungkan ke PostgreSQL sebagai sumber

Gunakan prosedur berikut untuk terhubung ke basis data sumber PostgreSQL Anda dengan basis data sumber AWS Schema Conversion Tool.

Untuk terhubung ke basis data sumber PostgreSQL

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih PostgreSQL, lalu pilih Berikutnya.

Kotak dialog Tambahkan sumber muncul.

3. Untuk nama Koneksi, masukkan nama untuk basis data Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensial database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensial database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Secret, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Secrets Manager.

Untuk informasi tentang penggunaan kredensial database dari Secrets Manager, lihat [Menggunakan AWS Secrets Manager](#).

- Untuk memasukkan informasi koneksi database sumber PostgreSQL secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama server	<p>Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.</p> <p>Anda dapat terhubung ke basis data PostgreSQL sumber Anda menggunakan protokol alamat IPv6. Untuk melakukannya, pastikan bahwa Anda menggunakan tanda kurung persegi untuk memasukkan alamat IP, seperti yang ditunjukkan dalam contoh berikut.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; text-align: center; margin-top: 10px;"> <p>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</p> </div>

Parameter	Action
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server basis data sumber Anda.
Basis data	Masukkan nama basis data PostgreSQL.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi basis data untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke basis data Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none"> • Verifikasi sertifikat server: Pilih opsi ini untuk memverifikasi sertifikat server menggunakan toko kepercayaan. • Toko kepercayaan: Lokasi toko perwalian yang berisi sertifikat. Agar lokasi ini muncul di bagian Pengaturan global, pastikan untuk menambahkannya.
Simpan kata sandi	<p>AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Mengaktifkan opsi ini memungkinkan Anda menyimpan password database dan untuk terhubung dengan cepat ke database tanpa harus memasukkan password.</p>

Parameter	Action
Jalur driver PostgreSQL	<p>Masukkan jalur ke pengantaran yang akan digunakan untuk terhubung ke basis data sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

- Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke basis data sumber Anda.
- Pilih Connect untuk terhubung ke basis data sumber Anda.

Hak istimewa untuk MySQL sebagai basis data target

Hak istimewa yang diperlukan untuk MySQL sebagai target saat Anda bermigrasi dari PostgreSQL adalah sebagai berikut:

- BUAT PADA* . *
- MENGUBAH DI* . *
- JATUHKAN DI* . *
- INDEKS DI* . *
- REFERENSI DI* . *
- PILIH PADA* . *
- BUAT TAMPILAN PADA* . *
- TAMPILKAN TAMPILAN DI* . *
- PEMICU DI* . *
- BUAT RUTINITAS DI* . *
- MENGUBAH RUTINITAS PADA* . *
- MENGEKSEKUSI PADA* . *
- INSERT, UPDATE PADA AWS_POSTGRESQL_EXT. *
- MENYISIPKAN, MEMPERBARUI, MENGHAPUS PADA AWS_POSTGRESQL_EXT_DATA. *

- MEMBUAT TABEL SEMENTARA PADA AWS_POSTGRESQL_EXT_DATA. *

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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';
```

Di contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *kata sandi Anda dengan kata* sandi yang aman.

Untuk menggunakan Amazon RDS for MySQL atau Aurora MySQL sebagai target, tetapkan `lower_case_table_names` parameternya1. Nilai ini berarti bahwa server MySQL menangani pengidentifikasi nama objek seperti tabel, indeks, pemicu, dan database sebagai kasus sensitif. Jika Anda telah mengaktifkan pencatatan biner di instance target Anda, maka atur `log_bin_trust_function_creators` parameternya1. Dalam hal ini, Anda tidak perlu menggunakan `DETERMINISTIC, READS SQL DATA` atau `NO SQL` karakteristik untuk membuat fungsi yang tersimpan. Untuk mengonfigurasi parameter ini, buat grup parameter DB baru atau ubah grup parameter DB yang ada.

Menggunakan SAP ASE (Sybase ASE) sebagai sumber AWS SCT

Anda dapat menggunakan AWS SCT untuk mengkonversi skema, objek kode database, dan kode aplikasi dari SAP (Sybase) Adaptive Server Enterprise (ASE) ke target berikut:

- Amazon RDS for MySQL

- Amazon Aurora Edisi Kompatibel MySQL
- Amazon RDS for MariaDB
- Amazon RDS for PostgreSQL
- Amazon Aurora Edisi Kompatibel PostgreSQL

Untuk informasi selengkapnya, lihat bagian berikut:

Topik

- [Keistimewaan untuk SAP ASE sebagai basis data sumber](#)
- [Menghubungkan ke SAP ASE \(Sybase\) sebagai sumber](#)
- [Hak istimewa untuk MySQL sebagai basis data target](#)
- [Pengaturan konversi SAP ASE ke MySQL](#)
- [Hak istimewa untuk PostgreSQL sebagai basis data target](#)
- [Pengaturan konversi SAP ASE ke PostgreSQL](#)

Keistimewaan untuk SAP ASE sebagai basis data sumber

Untuk menggunakan basis data dan memberikan izin. Untuk melakukannya, ikuti langkah-langkah berikut.

Buat dan konfigurasi pengguna basis data

1. Connect ke basis data sumber.
2. Buat pengguna database dengan perintah berikut. Berikan kata sandi untuk pengguna baru.

```
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. Untuk setiap database yang akan Anda migrasi, berikan hak istimewa berikut.

```
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
```

Menghubungkan ke SAP ASE (Sybase) sebagai sumber

Gunakan prosedur berikut untuk Connect ke basis data sumber SAP ASE Anda dengan AWS Schema Conversion Tool.

Untuk terhubung ke database sumber SAP ASE


1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih SAP ASE, lalu pilih Next.

Kotak dialog Tambahkan sumber akan muncul.

3. Untuk Nama Connection, masukkan nama untuk basis data Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensial database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Secret, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Secrets Manager.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat [Menggunakan AWS Secrets Manager](#).

- Untuk memasukkan informasi koneksi database sumber SAP ASE secara manual, gunakan instruksi berikut:

Parameter	Action
Nama server	Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
Pelabuhan server	Masukkan port yang digunakan untuk Connect ke server basis data sumber Anda.
Basis data	Masukkan nama basis data SAP ASE.
Nama pengguna dan Kata Sandi	Masukkan kredensial basis data sumber Anda. <div data-bbox="656 667 1510 1268" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;"><p> Note</p><p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p></div>
Gunakan SSL	Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke basis data Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL: <ul style="list-style-type: none">• Verifikasi sertifikat server: Pilih opsi ini untuk memverifikasi sertifikat server menggunakan toko kepercayaan.• Toko kepercayaan: Lokasi toko perwalian yang berisi sertifikat.

Parameter	Action
Simpan kata sandi	AWS SCT menciptakan brankas aman untuk menyimpan sertifikat SSL dan kata sandi database. Mengaktifkan opsi ini memungkinkan Anda menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.
Jalur driver SAP ASE	<p>Masukkan jalur ke basis data sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

- Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat Connect ke basis data sumber Anda.
- Pilih Connect untuk Connect ke basis data sumber Anda.

Hak istimewa untuk MySQL sebagai basis data target

Hak istimewa yang diperlukan untuk MySQL sebagai target adalah sebagai berikut:

- BUAT PADA* . *
- MENGUBAH DI * . *
- JATUHKAN . *
- INDEKS DI* . *
- REFERENSI DI * . *
- PILIH PADA* . *
- BUAT TAMPILAN PADA* . *
- TAMPILKAN TAMPILAN DI* . *
- PEMICU . *
- BUAT RUTINITAS DI* . *

- MENGUBAH RUTINITAS PADA* . *
- MENGEKSEKUSI PADA* . *
- INSERT, UPDATE PADA AWS_SAPASE_EXT. *
- MEMBUAT TABEL SEMENTARA PADA AWS_SAPASE_EXT. *

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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';
```

Di contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *kata sandi Anda dengan kata* sandi yang aman.

Untuk menggunakan Amazon RDS for MySQL atau Aurora MySQL sebagai target, tetapkan `lower_case_table_names` parameternya1. Nilai ini berarti bahwa server MySQL menangani pengidentifikasi nama objek seperti tabel, indeks, pemicu, dan database sebagai kasus sensitif. Jika Anda telah mengaktifkan pencatatan biner di instance target Anda, maka atur `log_bin_trust_function_creators` parameternya1. Dalam hal ini, Anda tidak perlu menggunakan `DETERMINISTIC, READS SQL DATA` atau `NO SQL` karakteristik untuk membuat fungsi yang tersimpan. Untuk mengkonfigurasi parameter ini, buat kelompok parameter DB baru atau ubah kelompok parameter DB yang sudah ada.

Pengaturan konversi SAP ASE ke MySQL

Untuk mengedit pengaturan konversi SAP ASE ke MySQL, pilih Pengaturan, lalu pilih Pengaturan konversi. Dari daftar atas, pilih SAP ASE, lalu pilih SAP ASE - MySQL atau SAP ASE - Amazon Aurora (kompatibel dengan MySQL). AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi SAP ASE ke PostgreSQL.

Pengaturan konversi SAP ASE ke MySQL AWS SCT termasuk opsi untuk berikut ini:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menggunakan nama yang tepat dari objek database sumber dalam kode dikonversi.

Secara default, AWS SCT mengkonversi nama-nama objek database, variabel, dan parameter untuk huruf kecil. Untuk menjaga kasus asli untuk nama-nama ini, pilih Perlakukan nama objek database sumber sebagai case sensitive. Pilih opsi ini jika Anda menggunakan nama objek case-sensitive di server database SAP ASE sumber Anda.

Hak istimewa untuk PostgreSQL sebagai basis data target

Untuk menggunakan PostgreSQL sebagai target, AWS SCT memerlukan `CREATE ON DATABASE` hak istimewa. Pastikan Anda memberikan hak istimewa ini untuk setiap database PostgreSQL target.

Untuk menggunakan sinonim publik yang dikonversi, ubah jalur pencarian default basis data ke `"$user", public_synonyms, public`.

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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;
```

Di contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *db_name* dengan nama database target Anda. Terakhir, ganti *kata sandi Anda dengan kata* sandi yang aman.

Di PostgreSQL, hanya pemilik skema `atauseruser` dapat menjatuhkan skema. Pemilik dapat menjatuhkan skema dan semua objek yang disertakan skema ini bahkan jika pemilik skema tidak memiliki beberapa objeknya.

Bila Anda menggunakan pengguna yang berbeda untuk mengkonversi dan menerapkan skema yang berbeda ke database target Anda, Anda bisa mendapatkan pesan kesalahan ketika tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan galat ini, gunakan `superuser` peran.

Pengaturan konversi SAP ASE ke PostgreSQL

Untuk mengedit pengaturan konversi SAP ASE ke PostgreSQL, pilih Pengaturan, lalu pilih Pengaturan konversi. Dari daftar atas, pilih SAP ASE, lalu pilih SAP ASE - PostgreSQL atau SAP ASE - Amazon Aurora (kompatibel dengan PostgreSQL). AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi SAP ASE ke PostgreSQL.

Pengaturan konversi SAP ASE ke PostgreSQL di AWS SCT termasuk opsi untuk berikut ini:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menentukan template yang akan digunakan untuk nama-nama skema dalam kode dikonversi. Untuk Templat pembuatan nama skema, pilih salah satu opsi berikut:
 - `<source_db>`- Menggunakan nama database SAP ASE sebagai nama skema di PostgreSQL.
 - `<source_schema>`- Menggunakan nama skema SAP ASE sebagai nama skema di PostgreSQL.

- `<source_db><schema>`- Menggunakan kombinasi database SAP ASE dan nama skema sebagai nama skema di PostgreSQL.
- Untuk menggunakan nama yang tepat dari objek database sumber dalam kode dikonversi.

Secara default,AWS SCT mengkonversi nama-nama objek database, variabel, dan parameter untuk huruf kecil. Untuk menjaga kasus asli untuk nama-nama ini, pilih Perlakukan nama objek database sumber sebagai case sensitive. Pilih opsi ini jika Anda menggunakan nama objek case-sensitive di server database SAP ASE sumber Anda.

Untuk operasi case-sensitive,AWS SCT dapat menghindari konversi nama objek database ke huruf kecil. Untuk melakukannya, pilih Hindari pengecoran ke huruf kecil untuk operasi sensitif huruf.

- Untuk memungkinkan penggunaan indeks dengan nama yang sama dalam tabel yang berbeda di SAP ASE.

Di PostgreSQL, semua nama indeks yang Anda gunakan dalam skema harus unik. Untuk memastikan bahwaAWS SCT menghasilkan nama unik untuk semua indeks Anda, pilih Buat nama unik untuk indeks.

Menggunakan Microsoft SQL Server sebagai sumber untukAWS SCT

Anda bisa menggunakanAWS SCTuntuk mengonversi skema, objek kode database, dan kode aplikasi dari SQL Server ke target berikut:

- Amazon RDS for MySQL
- Amazon Aurora Edisi Kompatibel MySQL
- Amazon RDS for PostgreSQL
- Amazon Aurora Edisi Kompatibel PostgreSQL
- Amazon RDS for SQL Server
- Amazon RDS for MariaDB

Note

AWS SCTtidak mendukung penggunaan Amazon RDS untuk SQL server sebagai sumber.

Anda bisa menggunakan AWS SCT untuk membuat laporan penilaian untuk migrasi skema, objek kode database, dan kode aplikasi dari SQL Server ke Babelfish untuk Aurora PostgreSQL, seperti yang dijelaskan berikut.

Topik

- [Hak istimewa untuk Microsoft SQL Server sebagai sumber](#)
- [Menggunakan Windows Authentication saat menggunakan Microsoft SQL Server sebagai sumber](#)
- [Menghubungkan ke SQL Server sebagai sumber](#)
- [Mengonversi SQL Server ke MySQL](#)
- [Mengkonversi SQL Server ke PostgreSQL](#)
- [Mengonversi SQL Server ke Amazon RDS for SQL Server](#)

Hak istimewa untuk Microsoft SQL Server sebagai sumber

Hak istimewa yang diperlukan untuk Microsoft SQL Server sebagai sumber adalah sebagai berikut:

- LIHAT DEFINISI
- ATUR DATABASE

The `VIEW DEFINITION` privilege memungkinkan pengguna yang memiliki akses publik untuk melihat definisi objek. AWS SCT menggunakan `VIEW DATABASE STATE` hak istimewa untuk memeriksa fitur edisi SQL Server Enterprise.

Ulangi hibah untuk setiap database yang skema Anda konversi.

Selain itu, berikan hak istimewa berikut pada master basis data:

- LIHAT STATUS SERVER
- LIHAT DEFINISI APA PUN

AWS SCT menggunakan `VIEW SERVER STATE` hak istimewa untuk mengumpulkan pengaturan dan konfigurasi server. Atur bahwa Anda memberikan `VIEW ANY DEFINITION` hak istimewa untuk melihat titik akhir.

Untuk membaca informasi tentang Microsoft Analysis Services, jalankan perintah berikut pada master basis data.

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

Dalam contoh selengkapnya tentang `<user_name>` placeholder dengan nama pengguna yang Anda berikan dengan hak istimewa sebelumnya.

Untuk membaca informasi tentang SQL Server Agent, tambahkan pengguna Anda ke `SQLAgentUser` peran. Jalankan perintah berikut pada `msdb` basis data.

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

Dalam contoh selengkapnya tentang `<SQLAgentRole>` placeholder dengan nama peran SQL Server Agent. Kemudian ganti `<user_name>` placeholder dengan nama pengguna yang Anda berikan dengan hak istimewa sebelumnya. Untuk informasi selengkapnya, lihat [Atur pengguna ke SQLAgentUser peranti](#) Panduan Pengguna Amazon RDS.

Untuk mendeteksi pengiriman log, berikan `SELECT` on `dbo.log_shipping_primary_databases` hak istimewa `msdb` basis data.

Untuk menggunakan pendekatan notifikasi replikasi DDL, berikan `RECEIVE ON <schema_name>.<queue_name>` hak istimewa pada basis data sumber Anda. Dalam contoh ini, ganti `<schema_name>` placeholder dengan nama skema database Anda. Kemudian, ganti `<queue_name>` placeholder dengan nama tabel antrian.

Menggunakan Windows Authentication saat menggunakan Microsoft SQL Server sebagai sumber

Jika aplikasi Anda berjalan pada intranet berbasis Windows, Anda mungkin dapat menggunakan Windows Authentication untuk akses database. Windows Authentication menggunakan identitas Windows saat ini yang ditetapkan pada thread sistem operasi untuk mengakses database SQL Server. Anda kemudian dapat memetakan identitas Windows ke database SQL Server dan izin. Untuk terhubung ke SQL Server menggunakan Windows Authentication, Anda harus menentukan identitas Windows yang digunakan aplikasi Anda. Anda juga harus memberikan akses identitas Windows ke database SQL Server.

SQL Server memiliki dua mode akses: mode Otentikasi Windows dan Mode Campuran. Mode Otentikasi Windows memungkinkan Otentikasi Windows dan menonaktifkan Otentikasi SQL Server. Mode Campuran memungkinkan Otentikasi Windows dan Otentikasi SQL Server. Otentikasi Windows selalu tersedia dan tidak dapat dinonaktifkan. Untuk informasi selengkapnya tentang Otentikasi Windows, lihat dokumentasi Microsoft Windows.

Contoh yang mungkin untuk membuat pengguna di TEST_DB ditampilkan sebagai berikut.

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

Menggunakan Otentikasi Windows dengan koneksi JDBC

Driver JDBC tidak mendukung Otentikasi Windows ketika driver digunakan pada sistem operasi non-Windows. Kredensial Windows Authentication, seperti nama pengguna dan kata sandi, tidak ditentukan secara otomatis saat menghubungkan ke SQL Server dari sistem operasi non-Windows. Dalam kasus seperti itu, aplikasi harus menggunakan SQL Server Authentication sebagai gantinya.

Dalam string koneksi JDBC, parameternya `integratedSecurity` harus ditentukan untuk terhubung menggunakan Windows Authentication. Driver JDBC mendukung Otentikasi Windows Terpadu pada sistem operasi Windows melalui `integratedSecurity` parameter string koneksi.

Untuk menggunakan otentikasi terintegrasi

1. Atur driver JDBC.
2. Salin `sqljdbc_auth.dll` ke direktori di jalur sistem Windows di komputer tempat driver JDBC diinstal.

The `sqljdbc_auth.dll` ini diinstal di lokasi berikut ini:

```
<direktori instalasi>\sqljdbc_<versi>\<bahasa>\otentikasi\
```

Ketika Anda mencoba untuk membuat koneksi ke database SQL Server menggunakan Windows Authentication, Anda mungkin mendapatkan kesalahan ini: Driver ini tidak dikonfigurasi untuk otentikasi terintegrasi. Masalah ini dapat diselesaikan dengan melakukan tindakan berikut:

- Deklarasikan dua variabel yang mengarah ke jalur terinstal JDBC Anda:

```
variable name: SQLJDBC_HOME; variable value: D:\lib\JDBC4.1\enu(di mana sqljdbc4.jar Anda ada);
```

```
variable name: SQLJDBC_AUTH_HOME; variable value: D\lib\JDBC4.1\enu
\auth\x86(jika Anda menjalankan OS 32bit) atauD\lib\JDBC4.1\enu\auth\x64(jika Anda
menjalankan OS 64bit). Di sinilah Anda sqljdbc_auth.dll terletak.
```

- Salin `sqljdbc_auth.dll` ke folder tempat JDK/JRE Anda berjalan. Anda dapat menyalin ke folder `lib`, folder `bin`, dan sebagainya. Sebagai contoh, Anda dapat menyalin ke folder berikut.

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

- Pastikan bahwa di folder perpustakaan JDBC Anda, Anda hanya memiliki file `SQLJDBC4.jar`. Hapus file `sqljdbc*.jar` lainnya dari folder itu (atau salin ke folder lain). Jika Anda menambahkan driver sebagai bagian dari program Anda, pastikan bahwa Anda hanya menambahkan `SQLJDBC4.jar` sebagai driver untuk digunakan.
- Salin `sqljdbc_auth.dll` file dalam folder dengan aplikasi Anda.

Note

Jika Anda menjalankan Java Virtual Machine (JVM) 32-bit, gunakan file `sqljdbc_auth.dll` di folder `x86`, bahkan jika sistem operasinya adalah versi `x64`. Jika Anda menjalankan JVM 64-bit pada prosesor `x64`, gunakan file `sqljdbc_auth.dll` di folder `x64`.

Ketika Anda terhubung ke database SQL Server, Anda dapat memilih salah satu Otentikasi Windows atau Otentikasi SQL Server untuk Otentikasi Pilihan.

Menghubungkan ke SQL Server sebagai sumber

Gunakan prosedur berikut untuk menyambung ke database sumber Microsoft SQL Server Anda dengan AWS Schema Conversion Tool.

Untuk terhubung ke database sumber Microsoft SQL Server

1. Di AWS Schema Conversion Tool, pilih **Tambahkan sumber**.
2. Pilih **Micro Microsoft SQL**, lalu pilih **Berikutnya**.

The **Tambahkan sumber** Atur kotak dialog.
3. Untuk **Nama koneksi**, masukkan nama untuk basis data Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensial basis data dari AWS Secrets Manager atau masukkan secara manual:

- Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk Rahasia AWS, pilih nama rahasianya.
 2. Pilih **Mengisi** untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Secrets Manager.

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat [Menggunakan AWS Secrets Manager](#).

- Untuk memasukkan informasi koneksi database sumber Microsoft SQL Server secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama server	<p>Masukkan nama Domain Name Services (DNS) atau alamat IP dari server database sumber Anda.</p> <p>Anda dapat terhubung ke database SQL Server Anda menggunakan protokol alamat IPv6. Untuk melakukannya, pastikan Anda menggunakan tanda kurung untuk memasukkan alamat IP, seperti yang ditunjukkan pada contoh berikut.</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>
Port server	Masukkan port yang digunakan untuk terhubung ke server basis data sumber Anda.
Nama	Masukkan nama instance untuk database SQL Server. Untuk menemukan nama instance, jalankan kueri <code>SELECT @@servername;</code> pada database SQL Server Anda.
Otentikasi	Pilih jenis otentikasi dari Otentikasi Windows dan Otentikasi SQL Server .

Parameter	Action
Nama tangandanKata Sandi	<p>Masukkan kredensial database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT memproyeksikan dan membukanya kembali, Anda diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada SSL tab:</p> <ul style="list-style-type: none">• Atur sertifikat server: Pilih opsi ini untuk mempercayai sertifikat server.• Toko kepercayaan: Lokasi toko kepercayaan yang berisi sertifikat. Agar lokasi ini muncul di Atur global bagian, pastikan untuk menambahkannya.
Menyimpan kata sandi	<p>AWS SCT membuat brankas aman untuk menyimpan sertifikat SSL dan kata sandi database. Mengaktifkan opsi ini memungkinkan Anda menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.</p>
Jalur Pengemudi Server Sql	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

Parameter	Action
Pustaka Otentikasi Windows	Masuki jalan menuju sqljdbc_auth.dll berkas. Secara default, file ini diinstal di lokasi berikut ini: <i><installation directory of the JDBC driver></i> sqljdbc_ <i><version></i> \<i>language</i> \auth\

- Pilih Uji Koneksi. Verifikasi bahwa AWS SCT dapat terhubung ke basis data sumber Anda.
- Pilih Connect untuk terhubung ke basis data sumber Anda.

Mengonversi SQL Server ke MySQL

Untuk meniru fungsi database Microsoft SQL Server dalam kode MySQL Anda yang dikonversi, gunakan paket ekstensi SQL Server ke MySQL AWS SCT. Untuk informasi selengkapnya tentang paket ekstensi ini, lihat [Menggunakan paket AWS SCT ekstensi](#).

Topik

- [Hak istimewa untuk MySQL sebagai basis data target](#)
- [SQL Server ke pengaturan konversi MySQL](#)
- [Pertimbangan migrasi](#)

Hak istimewa untuk MySQL sebagai basis data target

Hak istimewa yang diperlukan untuk MySQL sebagai target adalah sebagai berikut:

- BUAT DI *.*
- MENGUBAH PADA *.*
- JATUH PADA .*
- INDEKS PADA .*
- REFERENSI PADA *.*
- PILIH PADA *.*
- BUAT TAMPILAN DI *.*
- TAMPILKAN TAMPILAN DI *.*
- PEMILICU PADA *.*

- BUAT RUTINITAS DI *.*
- MENGUBAH RUTINITAS PADA *.*
- MENGEKSEKUSI PADA *.*
- SISIPKAN, PERBARUI PADA AWS_SQLSERVER_EXT.*
- MASUKKAN, PERBARUI, HAPUS PADA AWS_SQLSERVER_EXT_DATA.*
- BUAT TABEL SEMENTARA DI AWS_SQLSERVER_EXT_DATA.*

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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';
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *kata sandi_Anda* dengan kata sandi yang aman.

Jika Anda menggunakan database MySQL versi 5.7 atau lebih rendah sebagai target, maka jalankan perintah berikut. Untuk database MySQL versi 8.0 dan yang lebih tinggi, perintah ini tidak digunakan lagi.

```
GRANT SELECT ON mysql.proc TO 'user_name';
```

Untuk menggunakan Amazon RDS for MySQL atau Aurora MySQL sebagai target, tetapkan `lower_case_table_names` Parameter ke 1. Nilai ini berarti bahwa server MySQL

menangani pengidentifikasi nama objek seperti tabel, indeks, pemicu, dan database sebagai case insensitive. Jika Anda telah mengaktifkan logging biner di instance target Anda, maka atur `log_bin_trust_function_creators` Parameter ke 1. Dalam hal ini, Anda tidak perlu menggunakan `DETERMINISTIC, READS SQL DATA` atau `NO SQL` karakteristik untuk membuat fungsi yang disimpan. Untuk mengkonfigurasi parameter ini, buat grup parameter DB baru atau modifikasi grup parameter DB yang ada.

SQL Server ke pengaturan konversi MySQL

Untuk mengedit SQL Server ke pengaturan konversi MySQL, di AWS SCT pilih Pengaturan, dan kemudian pilih Atur konversi. Dari daftar atas, pilih Server SQL, dan kemudian pilih SQL Server - MySQL. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi SQL Server ke MySQL.

SQL Server ke pengaturan konversi MySQL di AWS SCT sertakan opsi untuk hal-hal berikut:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang dikonversi, pilih Semua pesan.

- Untuk memungkinkan database SQL Server sumber Anda untuk menyimpan output EXEC dalam sebuah meja. AWS SCT membuat tabel sementara dan prosedur tambahan untuk meniru fitur ini. Untuk menggunakan emulasi ini, pilih Buat rutinitas tambahan untuk menangani kumpulan data terbuka.

Pertimbangan migrasi

Pertimbangkan hal-hal ini saat memigrasikan skema SQL Server ke MySQL:

- MySQL tidak mendukung MERGE pernyataan. Namun, AWS SCT dapat meniru MERGE pernyataan selama konversi dengan menggunakan `INSERT ON DUPLICATE KEY` klausa dan `UPDATE FROM` and `DELETE FROM` pernyataan.

Untuk emulasi yang benar menggunakan `INSERT ON DUPLICATE KEY`, pastikan bahwa kendala unik atau kunci utama ada pada database MySQL target.

- Anda dapat menggunakan `GOTO` pernyataan dan label untuk mengubah urutan pernyataan dijalankan. Setiap pernyataan Transact-SQL yang mengikuti `GOTO` pernyataan dilewati, dan pemrosesan berlanjut pada label. Anda bisa menggunakan `GOTO` pernyataan dan label di mana saja dalam prosedur, batch, atau blok pernyataan. Anda juga bisa bersarang `GOTO` pernyataan.

MySQL tidak menggunakan `GOTO` pernyataan. Kapan AWS SCT mengkonversi kode yang berisi `GOTO` pernyataan, itu mengubah pernyataan untuk menggunakan `BEGIN...END` atau `LOOP...END LOOP` pernyataan. Anda dapat menemukan contoh bagaimana AWS SCT berubah `GOTO` pernyataan dalam tabel berikut.

Pernyataan SQL Server	Pernyataan MySQL
<pre> BEGIN statement1; GOTO label1; statement2; label1: Statement3; END </pre>	<pre> BEGIN label1: BEGIN statement1; LEAVE label1; statement2; END; Statement3; END </pre>
<pre> BEGIN statement1; label1: statement2; GOTO label1; statement3; </pre>	<pre> BEGIN statement1; label1: LOOP statement2; ITERATE label1; LEAVE label1; </pre>

Pernyataan SQL Server	Pernyataan MySQL
<pre>statement4; END</pre>	<pre>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 tidak mendukung fungsi bernilai tabel multistatement. AWS SCT mensimulasikan fungsi bernilai tabel selama konversi dengan membuat tabel sementara dan menulis ulang pernyataan untuk menggunakan tabel sementara ini.

Mengkonversi SQL Server ke PostgreSQL

Anda dapat menggunakan SQL Server untuk paket ekstensi PostgreSQL di AWS SCT. Paket ekstensi ini mengemulasi fungsi database SQL Server dalam kode PostgreSQL yang dikonversi. Gunakan paket ekstensi SQL Server ke PostgreSQL untuk meniru SQL Server Agent dan SQL Server Database Mail. Untuk informasi selengkapnya tentang paket ekstensi ini, lihat [Menggunakan paket AWS SCT ekstensi](#).

Topik

- [Hak istimewa untuk PostgreSQL sebagai basis data target](#)

- [SQL Server ke pengaturan konversi PostgreSQL](#)
- [Mengkonversi partisi SQL Server ke PostgreSQL versi 10 partisi](#)
- [Pertimbangan migrasi](#)
- [Menggunakan sebuahAWS SCTpaket ekstensi untuk meniru SQL Server Agent di PostgreSQL](#)
- [Menggunakan sebuahAWS SCTpaket ekstensi untuk meniru SQL Server Database Mail di PostgreSQL](#)

Hak istimewa untuk PostgreSQL sebagai basis data target

Untuk menggunakan PostgreSQL sebagai target,AWS SCTmembutuhkanCREATE ON DATABASEhak istimewa. Pastikan Anda memberikan hak istimewa ini untuk setiap database PostgreSQL target.

Untuk menggunakan sinonim publik yang dikonversi, ubah jalur pencarian default database ke"\$user", public_synonyms, public.

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan hak istimewa.

```
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;
```

Pada contoh sebelumnya, ganti*user_name*dengan nama pengguna Anda. Kemudian, ganti*db_name*Dengan nama database target ini. Akhirnya, ganti*kata sandi_Anda*dengan kata sandi yang aman.

Di PostgreSQL, hanya pemilik skema atausuperuserdapat menjatuhkan skema. Pemilik dapat menjatuhkan skema dan semua objek yang disertakan skema ini bahkan jika pemilik skema tidak memiliki beberapa objeknya.

Saat Anda menggunakan pengguna yang berbeda untuk mengonversi dan menerapkan skema yang berbeda ke basis data target Anda, Anda bisa mendapatkan pesan kesalahan saatAWS SCTtidak bisa menjatuhkan skema. Untuk menghindari pesan kesalahan ini, gunakansuperuserperan.

SQL Server ke pengaturan konversi PostgreSQL

Untuk mengedit pengaturan konversi SQL Server ke PostgreSQL, pilihPengaturan, dan kemudian pilihAtur konversi. Dari daftar atas, pilihServer SQL, dan kemudian pilihSQL Server -

PostgreSQL.AWS SCTmenampilkan semua pengaturan yang tersedia untuk konversi SQL Server ke PostgreSQL.

SQL Server ke pengaturan konversi PostgreSQL diAWS SCTsertakan opsi untuk hal-hal berikut:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

UntukTambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan.AWS SCTmenambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode yang dikonversi, pilihKesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang dikonversi, pilihSemua pesan.

- Untuk memungkinkan untuk menggunakan indeks dengan nama yang sama dalam tabel yang berbeda di SQL Server.

Di PostgreSQL, semua nama indeks yang Anda gunakan dalam skema, harus unik. Untuk memastikan bahwaAWS SCTmenghasilkan nama unik untuk semua indeks Anda, pilihHasilkan nama unik untuk indeks.

- Untuk mengkonversi prosedur SQL Server ke fungsi PostgreSQL.

PostgreSQL versi 10 dan sebelumnya tidak mendukung prosedur. Untuk pelanggan yang tidak terbiasa menggunakan prosedur di PostgreSQL,AWS SCTdapat mengubah prosedur menjadi fungsi. Untuk melakukannya, pilihUbah prosedur menjadi fungsi.

- Untuk meniru output dariEXECdalam sebuah meja.

Database SQL Server sumber Anda dapat menyimpan outputEXECdalam sebuah meja.AWS SCTmembuat tabel sementara dan prosedur tambahan untuk meniru fitur ini. Untuk menggunakan emulasi ini, pilihBuat rutinitas tambahan untuk menangani kumpulan data terbuka.

- Untuk menentukan template yang akan digunakan untuk nama skema dalam kode yang dikonversi. UntukTemplat pembuatan nama skema, pilih salah satu opsi berikut ini:
 - <source_db>— Menggunakan nama database SQL Server sebagai nama skema di PostgreSQL.
 - <source_schema>— Menggunakan nama skema SQL Server sebagai nama skema di PostgreSQL.
 - <source_db>_ <schema>— Menggunakan kombinasi database SQL Server dan nama skema sebagai nama skema di PostgreSQL.

- Untuk menyimpan huruf besar nama objek sumber Anda.

Untuk menghindari konversi nama objek ke huruf kecil, pilih **Hindari casting ke huruf kecil untuk operasi peka huruf besar/kecil**. Opsi ini hanya berlaku ketika Anda mengaktifkan opsi sensitivitas kasus di basis data target Anda.

- Untuk menyimpan nama parameter dari database sumber Anda.

Untuk menambahkan tanda kutip ganda ke nama parameter dalam kode yang dikonversi, pilih **Simpan nama parameter asli**.

Mengkonversi partisi SQL Server ke PostgreSQL versi 10 partisi

Saat Anda mengonversi database Microsoft SQL Server ke Amazon Aurora PostgreSQL Compatible Edition (Aurora PostgreSQL) atau Amazon Relational Database Service untuk PostgreSQL (Amazon RDS for PostgreSQL), perhatikan hal-hal berikut.

Di SQL Server, Anda membuat partisi dengan fungsi partisi. Saat mengonversi dari tabel partisi SQL Server ke tabel partisi PostgreSQL versi 10, perhatikan beberapa masalah potensial:

- SQL Server memungkinkan Anda untuk partisi tabel menggunakan kolom tanpa kendala NOT NULL. Dalam hal ini, semua nilai NULL pergi ke partisi paling kiri. PostgreSQL tidak mendukung nilai NULL untuk partisi RANGE.
- SQL Server memungkinkan Anda membuat kunci primer dan unik untuk tabel yang dipartisi. Untuk PostgreSQL, Anda membuat kunci primer atau unik untuk setiap partisi secara langsung. Dengan demikian, kendala PRIMARY atau UNIQUE KEY harus dihapus dari tabel induknya saat bermigrasi ke PostgreSQL. Nama kunci yang dihasilkan mengambil format `<original_key_name>_<partition_number>`.
- SQL Server memungkinkan Anda membuat batasan kunci asing dari dan ke tabel yang dipartisi. PostgreSQL tidak mendukung kunci asing yang mereferensikan tabel yang dipartisi. Juga, PostgreSQL tidak mendukung referensi kunci asing dari tabel yang dipartisi ke tabel lain.
- SQL Server memungkinkan Anda membuat indeks untuk tabel yang dipartisi. Untuk PostgreSQL, indeks harus dibuat untuk setiap partisi secara langsung. Dengan demikian, indeks harus dihapus dari tabel induknya saat bermigrasi ke PostgreSQL. Nama indeks yang dihasilkan mengambil format `<original_index_name>_<partition_number>`.
- PostgreSQL tidak mendukung indeks yang dipartisi.

Pertimbangan migrasi

Beberapa hal yang perlu dipertimbangkan saat memigrasikan skema SQL Server ke PostgreSQL:

- Dalam PostgreSQL, semua nama objek dalam skema harus unik, termasuk indeks. Nama indeks harus unik dalam skema tabel dasar. Di SQL Server, nama indeks bisa sama untuk tabel yang berbeda.

Untuk memastikan keunikan nama indeks, AWS SCT memberi Anda pilihan untuk menghasilkan nama indeks unik jika nama indeks Anda tidak unik. Untuk melakukan ini, pilih opsi Atur nama indeks yang unik dalam properti proyek. Secara default, opsi ini diaktifkan. Jika opsi ini diaktifkan, nama indeks unik dibuat menggunakan format `IX_TABLE_NAME_INDEX_NAME`. Jika opsi ini dinonaktifkan, nama indeks tidak diubah.

- Pernyataan GOTO dan label dapat digunakan untuk mengubah urutan pernyataan dijalankan. Setiap pernyataan Transact-SQL yang mengikuti pernyataan GOTO dilewati dan pemrosesan berlanjut pada label. Pernyataan dan label GOTO dapat digunakan di mana saja dalam prosedur, batch, atau blok pernyataan. Pernyataan GOTO juga dapat disarangkan.

PostgreSQL tidak menggunakan pernyataan GOTO. Kapan AWS SCT mengkonversi kode yang berisi pernyataan GOTO, itu mengubah pernyataan untuk menggunakan pernyataan `BEGIN... END` atau `LOOP... END LOOP`. Anda dapat menemukan contoh bagaimana AWS SCT mengkonversi pernyataan GOTO dalam tabel berikut.

Pernyataan SQL Server GOTO dan pernyataan PostgreSQL dikonversi

Pernyataan SQL Server	Pernyataan PostgreSQL
<pre>BEGIN statement1; GOTO label1; statement2; label1: Statement3; END</pre>	<pre>BEGIN label1: BEGIN statement1; EXIT label1; statement2; END; Statement3; </pre>

Pernyataan SQL Server	Pernyataan PostgreSQL
<pre> BEGIN statement1; label1: statement2; GOTO label1; statement3; statement4; END </pre>	<pre> END 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 tidak mendukung pernyataan MERGE. AWS SCT mengemulasi perilaku pernyataan MERGE dengan cara berikut ini:
 - Dengan INSERT ON CONFLICT konstruksi.

- Dengan menggunakan pernyataan UPDATE FROM DHTML, seperti MERGE tanpa klausa WHEN NOT MATCHED.
- Dengan menggunakan CURSOR, seperti dengan klausa MERGE dengan DELETE atau dengan menggunakan pernyataan kondisi MERGE ON yang kompleks.
- AWS SCT dapat menambahkan pemicu database ke pohon objek saat Amazon RDS adalah targetnya.
- AWS SCT dapat menambahkan pemicu tingkat server ke pohon objek saat Amazon RDS adalah targetnya.
- SQL Server secara otomatis membuat dan mengelola deleted dan inserted tabel. Anda dapat menggunakan tabel sementara, memori-residen ini untuk menguji efek modifikasi data tertentu dan untuk mengatur kondisi untuk tindakan pemicu DML. AWS SCT dapat mengonversi penggunaan tabel ini di dalam pernyataan pemicu DML.
- AWS SCT dapat menambahkan server tertaut ke pohon objek saat Amazon RDS adalah targetnya.
- Saat bermigrasi dari Microsoft SQL Server ke PostgreSQL, fungsi SUSER_SNAME bawaan dikonversi sebagai berikut:
 - SUSER_SNAME — Mengembalikan nama login yang terkait dengan nomor identifikasi keamanan (SID).
 - SUSER_SNAME (<server_user_sid>) - Tidak didukung.
 - SUSER_SNAME () CURRENT_USER — Mengembalikan nama pengguna dari konteks eksekusi saat ini.
 - SUSER_SNAME (NULL) - Mengembalikan NULL.
- Mengonversi fungsi bernilai tabel didukung. Fungsi bernilai tabel mengembalikan tabel dan dapat menggantikan tabel dalam kueri.
- PATINDEX mengembalikan posisi awal kemunculan pertama pola dalam ekspresi tertentu pada semua tipe data teks dan karakter yang valid. Ia mengembalikan nol jika pola tidak ditemukan. Saat mengonversi dari SQL Server ke Amazon RDS for PostgreSQL, AWS SCT <pattern character><expression character varying> menggantikan kode aplikasi yang menggunakan PATINDEX dengan aws_sqlserver_ext.patindex (,).
- Di SQL Server, tipe tabel yang ditentukan pengguna adalah tipe yang mewakili definisi struktur tabel. Anda menggunakan tipe tabel yang ditentukan pengguna untuk mendeklarasikan parameter nilai tabel untuk prosedur atau fungsi yang disimpan. Anda juga dapat menggunakan tipe tabel yang ditentukan pengguna untuk mendeklarasikan variabel tabel yang ingin Anda gunakan dalam batch atau dalam badan prosedur atau fungsi yang disimpan. AWS SCT meniru jenis ini di PostgreSQL dengan membuat tabel sementara.

Saat mengonversi dari SQL Server ke PostgreSQL, AWS SCT mengkonversi objek sistem SQL Server menjadi objek dikenali di PostgreSQL. Tabel berikut ini menunjukkan bagaimana objek sistem ini dikonversi.

Kasus penggunaan MS SQL Server	Substitusi PostgreSQL
SYS.SKEMA	AWS_SQLSERVER_EXT.SYS_SKEMA
SYS.TABLES	AWS_SQLSERVER_EXT.SYS_TABLES
SYS.PANDANGAN	AWS_SQLSERVER_EXT.SYS_VIEWS
SYS.ALL_VIEWS	AWS_SQLSERVER_EXT.SYS_ALL_VIEWS
SYS.JENIS	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_KENDALA	AWS_SQLSERVER_EXT.SYS_KEY_KENDALA
SYS.IDENTITY_COLUMNS	AWS_SQLSERVER_EXT.SYS_IDENTITY_COLUMNS
SYS.PROSEDUR	AWS_SQLSERVER_EXT.SYS_PROCEDURES
SYS.INDEXES	AWS_SQLSERVER_EXT.SYS_INDEXES
SYS.SYSINDEX	AWS_SQLSERVER_EXT.SYS_SYSINDEXES
SYS.OBJECTS	AWS_SQLSERVER_EXT.SYS_OBJECTS

Kasus penggunaan MS SQL Server	Substitusi PostgreSQL
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.DATABASE	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

Kasus penggunaan MS SQL Server	Substitusi PostgreSQL
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

Menggunakan sebuah AWS SCT paket ekstensi untuk meniru SQL Server Agent di PostgreSQL

SQL Server Agent adalah layanan Microsoft Windows yang menjalankan pekerjaan SQL Server. SQL Server Agent menjalankan pekerjaan pada jadwal, dalam menanggapi acara tertentu, atau sesuai permintaan. Untuk informasi selengkapnya tentang Agen Server SQL, lihat [Dokumentasi teknis Microsoft](#).

PostgreSQL tidak memiliki konten yang setara dengan SQL Server Agent. Untuk meniru fitur SQL Server Agent, AWS SCT membuat paket ekstensi. Paket ekstensi ini menggunakan AWS Lambda dan Amazon CloudWatch. AWS Lambda mengimplementasikan antarmuka yang Anda gunakan untuk mengelola jadwal dan menjalankan pekerjaan. Amazon CloudWatch mempertahankan aturan jadwal.

AWS Lambda dan Amazon CloudWatch menggunakan parameter JSON untuk berinteraksi. Parameter JSON ini memiliki struktur berikut ini.

```
{
  "mode": mode,
  "parameters": {
    list of parameters
  },
  "callback": procedure name
}
```

Dalam contoh sebelumnya, *mode* adalah jenis tugas dan *list of parameters* adalah seperangkat parameter yang bergantung pada jenis tugas ini. Juga, *procedure name* adalah nama prosedur yang berjalan setelah tugas selesai.

AWS SCT menggunakan satu fungsi Lambda untuk mengontrol dan menjalankan pekerjaan. The CloudWatch aturan memulai menjalankan pekerjaan dan memberikan informasi yang diperlukan untuk memulai pekerjaan. Ketika CloudWatch pemicu aturan, itu memulai fungsi Lambda menggunakan parameter dari aturan.

Untuk membuat pekerjaan sederhana yang memanggil prosedur, gunakan format berikut.

```
{
  "mode": "run_job",
  "parameters": {
    "vendor": "mysql",
    "cmd": "lambda_db.nightly_job"
  }
}
```

Untuk membuat pekerjaan dengan beberapa langkah ini, gunakan format berikut ini, gunakan format berikut ini.

```
{
  "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  
    },  
    ...  
]  
}
```

Untuk meniru perilaku Agen SQL Server di PostgreSQL, AWS SCT paket ekstensi juga membuat tabel dan prosedur berikut.

Tabel yang meniru SQL Server Agent di PostgreSQL

Untuk meniru SQL Server Agent, paket ekstensi menggunakan tabel berikut:

sysjobs

Menyimpan informasi tentang pekerjaan.

sysjobsteps

Menyimpan informasi tentang langkah-langkah pekerjaan.

sysschedules

Menyimpan informasi tentang jadwal pekerjaan.

jadwal sysjob

Menyimpan informasi jadwal untuk pekerjaan individu.

sysjobhistory

Menyimpan informasi tentang menjalankan pekerjaan terjadwal.

Prosedur yang meniru SQL Server Agent di PostgreSQL

Untuk meniru SQL Server Agent, paket ekstensi menggunakan prosedur berikut:

`sp_add_pekerjaan`

Atur pekerjaan baru.

`sp_add_jobstep`

Atur langkah ke pekerjaan.

`sp_add_schedule`

Membuat aturan jadwal baru di Amazon CloudWatch. Anda dapat menggunakan jadwal ini dengan sejumlah pekerjaan.

`sp_attach_schedule`

Menetapkan jadwal untuk pekerjaan yang dipilih.

`sp_add_jobschedule`

Membuat aturan jadwal untuk pekerjaan di Amazon CloudWatch dan menetapkan target untuk aturan ini.

`sp_update_job`

Memperbarui atribut pekerjaan yang dibuat sebelumnya.

`sp_update_jobstep`

Memperbarui atribut langkah dalam pekerjaan.

`sp_update_schedule`

Memperbarui atribut aturan jadwal di Amazon CloudWatch.

`sp_update_jobschedule`

Memperbarui atribut jadwal untuk pekerjaan yang ditentukan.

`sp_delete_job`

Menghapus pekerjaan.

`sp_delete_jobstep`

Menghapus langkah pekerjaan dari pekerjaan.

sp_delete_schedule

Atur jadwal.

sp_delete_jobschedule

Menghapus aturan jadwal untuk pekerjaan yang ditentukan dari Amazon CloudWatch.

sp_detach_schedule

Menghapus hubungan antara jadwal dan pekerjaan.

get_jobs, update_job

Prosedur internal yang berinteraksi dengan AWS Elastic Beanstalk.

sp_verify_job_date, sp_verify_job_time, sp_verify_job, sp_verify_jobstep, sp_verify_job_identifiers, sp_verify_schedule_identifiers

Prosedur internal yang memeriksa pengaturan.

Sintaks untuk prosedur yang meniru SQL Server Agent di PostgreSQL

Theaws_sqlserver_ext.sp_add_jobprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_add_jobprosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber ini, lihat [Dokumentasi teknis 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
```


Theaws_sqlserver_ext.sp_add_jobstepprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_add_jobstepprosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber ini, lihat [Dokumentasi teknis 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_cmdexec_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
```

Theaws_sqlserver_ext.sp_add_scheduleprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_add_scheduleprosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber ini, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sp_attach_scheduleprosedur dalam paket ekstensi mengemulasimdb.dbo.sp_attach_scheduleprosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber ini, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sp_add_jobscheduleprosedur dalam paket ekstensi mengemulasimdb.dbo.sp_add_jobscheduleprosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber ini, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sp_delete_jobprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_delete_jobprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Agent, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sp_delete_jobstepprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_delete_jobstepprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Agent, lihat [Dokumentasi teknis Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_step_id integer = NULL::integer,  
out returncode integer
```

Theaws_sqlserver_ext.sp_delete_jobscheduleprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_delete_jobscheduleprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Agent, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sp_delete_scheduleprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_delete_scheduleprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Agent, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sp_detach_scheduleprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_detach_scheduleprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Agent, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sp_update_jobprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_update_jobprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Agent, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sp_update_jobscheduleprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_update_jobscheduleprosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber ini, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sp_update_jobstepprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_update_jobstepprosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber ini, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sp_update_scheduleprosedur dalam paket ekstensi mengemulasimssdb.dbo.sp_update_scheduleprosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber ini, lihat[Dokumentasi teknis 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
```

Contoh untuk menggunakan prosedur yang meniru SQL Server Agent di PostgreSQL

Untuk menambahkan pekerjaan baru, gunakanaws_sqlserver_ext.sp_add_jobprosedur seperti yang ditunjukkan berikut ini.

```
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');
```

Untuk menambahkan langkah pekerjaan baru, gunakanaws_sqlserver_ext.sp_add_jobstepprosedur seperti yang ditunjukkan berikut.

```
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');
```

Untuk menambahkan jadwal sederhana, gunakan `aws_sqlserver_ext.sp_add_schedule` prosedur seperti yang ditunjukkan berikut.

```
SELECT * FROM aws_sqlserver_ext.sp_add_schedule(  
  par_schedule_name := 'RunOnce',  
  par_freq_type := 1,  
  par_active_start_time := 233000);
```

Untuk mengatur jadwal pekerjaan, gunakan `aws_sqlserver_ext.sp_attach_schedule` prosedur seperti yang ditunjukkan berikut.

```
SELECT * FROM aws_sqlserver_ext.sp_attach_schedule (  
  par_job_name := 'test_job',  
  par_schedule_name := 'NightlyJobs');
```

Untuk membuat jadwal pekerjaan, gunakan `aws_sqlserver_ext.sp_add_jobschedule` prosedur seperti yang ditunjukkan berikut.

```
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);
```

Gunakan contoh kasus untuk meniru Agen SQL Server di PostgreSQL

Jika kode database sumber Anda menggunakan SQL Server Agent untuk menjalankan pekerjaan, Anda dapat menggunakan paket ekstensi SQL Server ke PostgreSQLAWS SCT untuk mengonversi kode ini ke PostgreSQL. Paket ekstensi menggunakanAWS Lambda berfungsi untuk meniru perilaku SQL Server Agent.

Anda dapat membuat yang baruAWS Lambda fungsi atau mendaftarkan fungsi yang ada.

Untuk membuatAWS Lambda fungsi

1. DiAWS SCT, di pohon basis data target, buka menu konteks (klik kanan), pilih Terapkan paket ekstensi untuk, dan kemudian pilih PostgreSQL.

Wisaya paket ekstensi muncul.

2. Pada Layanan emulasi Agen SQL Server tab, lakukan hal berikut ini:

- Pilih Buat sebuahAWS Lambda fungsi.
- Untuk Login basis data, masukkan nama pengguna database target ini.
- Untuk Kata sandi basis data, masukkan kata sandi untuk nama pengguna yang Anda masukkan pada langkah sebelumnya.
- Untuk Folder pustaka Python, masukkan jalur ke folder perpustakaan Python Anda.
- Pilih BuatAWS Lambda fungsi, dan kemudian pilih Berikutnya.

Untuk mendaftarkanAWS Lambda fungsi yang Anda gunakan sebelumnya

- Jalankan skrip berikut ini pada basis data target Anda.

```
SELECT
  FROM aws_sqlserver_ext.set_service_setting(
    p_service := 'JOB',
    p_setting := 'LAMBDA_ARN',
    p_value := ARN)
```

Dalam contoh sebelumnya, *ARN* adalah Nama Sumber Daya Amazon (ARN) yang digunakanAWS Lambda fungsi.

Contoh berikut menciptakan tugas sederhana yang terdiri dari satu langkah. Setiap lima menit, tugas ini menjalankan yang sebelumnya dibuat `job_example` fungsi. Fungsi ini menyisipkan catatan ke `job_example_table`.

Untuk membuat tugas sederhana ini

1. Atur pekerjaan dengan menggunakan `aws_sqlserver_ext.sp_add_job` fungsi seperti yang ditunjukkan berikut ini.

```
SELECT
FROM aws_sqlserver_ext.sp_add_job (
    par_job_name := 'test_simple_job');
```

2. Atur langkah pekerjaan menggunakan `aws_sqlserver_ext.sp_add_jobstep` fungsi seperti yang ditunjukkan berikut ini.

```
SELECT
FROM aws_sqlserver_ext.sp_add_jobstep (
    par_job_name := 'test_simple_job',
    par_step_name := 'test_simple_job_step1',
    par_command := 'PERFORM job_simple_example;');
```

Langkah pekerjaan menentukan apa fungsi yang dilakukan.

3. Buat penjadwal untuk pekerjaan menggunakan `aws_sqlserver_ext.sp_add_jobschedule` fungsi seperti yang ditunjukkan berikut ini.

```
SELECT
FROM aws_sqlserver_ext.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 */);
```

Langkah pekerjaan menentukan apa fungsi yang dilakukan.

Untuk menghapus pekerjaan ini, gunakan `aws_sqlserver_ext.sp_delete_job` berfungsi seperti yang ditunjukkan berikut ini.

```
PERFORM aws_sqlserver_ext.sp_delete_job(  
    par_job_name := 'PeriodicJob1'::character varying,  
    par_delete_history := 1::smallint,  
    par_delete_unused_schedule := 1::smallint);
```

Menggunakan sebuah AWS SCT paket ekstensi untuk meniru SQL Server Database Mail di PostgreSQL

Anda dapat menggunakan SQL Server Database Mail untuk mengirim pesan email ke pengguna dari SQL Server Database Mail atau Azure SQL Managed Instance. Pesan email ini dapat berisi hasil kueri atau menyertakan file dari sumber daya apa pun di jaringan Anda. Untuk informasi selengkapnya tentang SQL Server Database Mail, lihat [Dokumentasi teknis Microsoft](#).

PostgreSQL tidak memiliki padanan untuk SQL Server Database Mail. Untuk meniru fitur SQL Server Database Mail, AWS SCT membuat paket ekstensi. Paket ekstensi ini menggunakan AWS Lambda dan Layanan Email Sederhana Amazon (Amazon SES). AWS Lambda menyediakan pengguna dengan antarmuka untuk berinteraksi dengan layanan pengiriman email Amazon SES. Untuk mengatur interaksi ini, tambahkan Amazon Resource Name (ARN) dari fungsi Lambda Anda.

Untuk akun email baru, gunakan perintah berikut.

```
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;
```

Untuk menambahkan ARN fungsi Lambda Anda ke akun email yang ada, gunakan perintah berikut.

```
do
```

```
$$
begin
PERFORM sysmail_update_account_sp (
    par_account_name := 'existind_account_name',
    par_mailserver_type := 'AWSLAMBDA'
    par_mailserver_name := 'ARN'
);
end;
$$ language plpgsql;
```

Dalam contoh-contoh sebelumnya, **ARN** adalah ARN dari fungsi Lambda Anda.

Untuk meniru perilaku SQL Server Database Mail di PostgreSQL, AWS SCT paket ekstensi menggunakan tabel, tampilan, dan prosedur berikut.

Tabel yang meniru SQL Server Database Mail di PostgreSQL

Untuk meniru SQL Server Database Mail, paket ekstensi menggunakan tabel berikut:

`sysmail_akun`

Menyimpan informasi selengkapnya tentang akun email ini.

`sysmail_profil`

Menyimpan informasi tentang profil pengguna.

`sysmail_server`

Menyimpan informasi selengkapnya tentang server email ini.

`sysmail_mailitems`

Menyimpan daftar pesan email ini.

`lampiran sysmail_`

Berkonten satu baris untuk setiap lampiran email ini.

`sysmail_log`

Menyimpan informasi layanan tentang mengirim pesan email.

`sysmail_profileakun`

Menyimpan informasi tentang profil pengguna dan akun email.

Tampilan yang meniru SQL Server Database Mail di PostgreSQL

Untuk meniru SQL Server Database Mail, AWS SCT membuat tampilan berikut dalam database PostgreSQL untuk memastikan kompatibilitas. Paket ekstensi tidak menggunakannya, tetapi kode yang dikonversi dapat menanyakan tampilan ini.

sysmail_allitems

Termasuk daftar semua email.

sysmail_faileditems

Termasuk daftar email yang tidak dapat dikirim.

sysmail_sentitems

Termasuk daftar email yang dikirim.

sysmail_unsentitems

Termasuk daftar email yang belum dikirim.

sysmail_mailattachments

Termasuk daftar file terlampir.

Prosedur yang meniru SQL Server Database Mail di PostgreSQL

Untuk meniru SQL Server Database Mail, paket ekstensi menggunakan prosedur berikut:

sp_send_dbmail

Mengirim email ke penerima yang ditentukan.

sysmail_add_profile_sp

Atur profil pengguna baru.

sysmail_add_account_sp

Membuat akun email baru yang menyimpan informasi seperti kredensial Simple Mail Transfer Protocol (SMTP), dan sebagainya.

sysmail_add_profileaccount_sp

Menambahkan akun email ke profil pengguna yang ditentukan.

sysmail_update_profile_sp

Mengubah atribut profil pengguna seperti deskripsi, nama, dan sebagainya.

sysmail_update_account_sp

Mengubah informasi di akun email yang ada.

sysmail_update_profileaccount_sp

Memperbarui informasi akun email di profil pengguna yang ditentukan.

sysmail_delete_profileaccount_sp

Menghapus akun email dari profil pengguna yang ditentukan.

sysmail_delete_account_sp

Menghapus akun email.

sysmail_delete_profile_sp

Atur profil pengguna.

sysmail_delete_mailitems_sp

Menghapus email dari tabel internal.

sysmail_help_profile_sp

Menampilkan informasi tentang profil pengguna.

sysmail_help_account_sp

Menampilkan informasi selengkapnya tentang akun email ini.

sysmail_help_profileaccount_sp

Menampilkan informasi tentang akun email yang terkait dengan profil pengguna.

sysmail_dbmail_json

Prosedur internal yang menghasilkan permintaan JSON untuk AWS Lambda fungsi.

sysmail_verify_profile_sp, sysmail_verify_account_sp, sysmail_verify_addressparams_sp

Prosedur internal yang memeriksa pengaturan.

sp_get_dbmail, sp_set_dbmail, sysmail_dbmail_xml

Prosedur internal yang tidak digunakan lagi.

Sintaks untuk prosedur yang meniru SQL Server Database Mail di PostgreSQL

Theaws_sqlserver_ext.sp_send_dbmailprosedur dalam paket ekstensi mengemulasimdb.dbo.sp_send_dbmailprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat[Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sysmail_delete_mailitems_spprosedur dalam paket ekstensi mengemulasimdb.dbo.sysmail_delete_mailitems_spprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat[Dokumentasi teknis Microsoft](#).

```
par_sent_before timestamp = NULL::timestamp without time zone,  
par_sent_status varchar = NULL::character varying,
```

```
out returncode integer
```

Theaws_sqlserver_ext.sysmail_add_profile_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_add_profile_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis Microsoft](#).

```
par_profile_name varchar,  
par_description varchar = NULL::character varying,  
out par_profile_id integer,  
out returncode integer
```

Theaws_sqlserver_ext.sysmail_add_account_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_add_account_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sysmail_add_profileaccount_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_add_profileaccount_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sysmail_help_profile_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_help_profile_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
out returncode integer
```

Theaws_sqlserver_ext.sysmail_update_profile_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_update_profile_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_description varchar = NULL::character varying,  
out returncode integer
```

Theaws_sqlserver_ext.sysmail_delete_profile_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_delete_profile_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_force_delete smallint = 1,  
out returncode integer
```

Theaws_sqlserver_ext.sysmail_help_account_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_help_account_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

Theaws_sqlserver_ext.sysmail_update_account_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_update_account_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sysmail_delete_account_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_delete_account_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

Theaws_sqlserver_ext.sysmail_help_profileaccount_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_help_profileaccount_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis 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
```

Theaws_sqlserver_ext.sysmail_update_profileaccount_sprosedur dalam paket ekstensi mengemulasimssdb.dbo.sysmail_update_profileaccount_sprosedur. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis 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
```

The `aws_sqlserver_ext.sysmail_delete_profileaccount_sprosedur` dalam paket ekstensi mengemulas `msdb.dbo.sysmail_delete_profileaccount_sprosedur`. Untuk informasi selengkapnya tentang prosedur SQL Server Database Mail, lihat [Dokumentasi teknis 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
```

Contoh untuk menggunakan prosedur yang meniru SQL Server Database Mail di PostgreSQL

Untuk mengirim email, gunakan `aws_sqlserver_ext.sp_send_dbmail` prosedur seperti yang ditunjukkan berikut.

```
PERFORM sp_send_dbmail (  
    par_profile_name := 'Administrator',  
    par_recipients := 'hello@rusgl.info',  
    par_subject := 'Automated Success Message',  
    par_body := 'The stored procedure finished'  
);
```

Contoh berikut ini menunjukkan cara mengirim email dengan hasil kueri.

```
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'  
);
```

Contoh berikut ini menunjukkan cara mengirim email dengan kode HTML.

```
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'
);
```

Untuk menghapus email,

gunakan `aws_sqlserver_ext.sysmail_delete_mailitems_sprosedur` seperti yang ditunjukkan berikut.

```
DECLARE var_GETDATE datetime;
SET var_GETDATE = NOW();
PERFORM sysmail_delete_mailitems_sp (
  par_sent_before := var_GETDATE
);
```

Contoh berikut ini menunjukkan cara menghapus email tertua.

```
PERFORM sysmail_delete_mailitems_sp (
  par_sent_before := '31.12.2015'
);
```

Contoh berikut menunjukkan cara menghapus semua email yang tidak dapat dikirim.

```
PERFORM sysmail_delete_mailitems_sp (
  par_sent_status := 'failed'
);
```

Untuk membuat profil pengguna baru ini,

gunakan `aws_sqlserver_ext.sysmail_add_profile_sprosedur` seperti yang ditunjukkan berikut.

```
PERFORM sysmail_add_profile_sp (  
    profile_name := 'Administrator',  
    par_description := 'administrative mail'  
);
```

Contoh berikut menunjukkan cara membuat profil baru dan menyimpan pengenal profil unik dalam variabel.

```
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;
```

Untuk membuat akun email ini, gunakan `aws_sqlserver_ext.sysmail_add_account_sprosedur` seperti yang ditunjukkan berikut.

```
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'  
);
```

Untuk menambahkan akun email ke profil pengguna, gunakan `aws_sqlserver_ext.sysmail_add_profileaccount_sprosedur` seperti yang ditunjukkan berikut.

```
PERFORM sysmail_add_profileaccount_sp (  
    par_account_name := 'Administrator',  
    par_account_name := 'Audit Account',  
    par_sequence_number := 1  
);
```

Gunakan contoh kasus untuk meniru SQL Server Database Mail di PostgreSQL

Jika kode basis data sumber Anda menggunakan SQL Server Database Mail untuk mengirim email, Anda dapat menggunakan AWS SCT paket ekstensi untuk mengonversi kode ini ke PostgreSQL.

Untuk mengirim email dari database PostgreSQL Anda

1. Atur dan Atur AWS Lambda fungsi.
2. Terapkan AWS SCT paket ekstensi.
3. Buat profil pengguna menggunakan `sysmail_add_profile_sp` berfungsi seperti yang ditunjukkan berikut ini.
4. Buat akun email menggunakan `sysmail_add_account_sp` berfungsi seperti yang ditunjukkan berikut ini.
5. Tambahkan akun email ini ke profil pengguna Anda menggunakan `sysmail_add_profileaccount_sp` berfungsi seperti yang ditunjukkan berikut ini.

```
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. Kirim email menggunakan `sp_send_dbmail` berfungsi seperti yang ditunjukkan berikut ini.

```
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;
```

Untuk melihat informasi tentang semua profil pengguna, gunakan `sysmail_help_profile_sp` seperti yang ditunjukkan berikut.

```
SELECT FROM aws_sqlserver_ext.sysmail_help_profile_sp();
```

Contoh berikut menampilkan informasi tentang profil pengguna tertentu.

```
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');
```

Untuk melihat informasi tentang semua akun email, gunakan `sysmail_help_account_sp` seperti yang ditunjukkan berikut.

```
select from aws_sqlserver_ext.sysmail_help_account_sp();
```

Contoh berikut menampilkan informasi tentang akun email tertentu.

```
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');
```

Untuk melihat informasi tentang semua akun email yang terkait dengan profil pengguna, gunakan `sysmail_help_profileaccount_sp` prosedur seperti yang ditunjukkan berikut.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp();
```

Contoh berikut memfilter catatan berdasarkan pengenalan, nama profil, atau nama akun.

```
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');
```

Untuk mengubah nama atau deskripsi profil pengguna, gunakan `sysmail_update_profile_sp` prosedur seperti yang ditunjukkan berikut.

```
select aws_sqlserver_ext.sysmail_update_profile_sp(
par_profile_id := 2,
par_profile_name := 'New profile name'
);
```

Untuk mengubah pengaturan akun email, gunakan `sysmail_update_account_sp` prosedur seperti yang ditunjukkan berikut.

```
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'
);
```

Mengonversi SQL Server ke Amazon RDS for SQL Server

Beberapa hal yang perlu dipertimbangkan saat memigrasikan skema dan kode SQL Server ke Amazon RDS for SQL Server:

- AWS SCT dapat mengonversi SQL Server Agent untuk memberikan jadwal, peringatan, dan pekerjaan di Amazon RDS for SQL Server DB. Setelah konversi, Anda dapat menggunakan instans Amazon RDS for SQL Server DB dengan SQL Server Reporting Services (SSRS), SQL Server Analysis Services (SSAS), dan SQL Server Integration Services (SSIS).
- Amazon RDS saat ini tidak mendukung Broker Layanan SQL Server atau titik akhir T-SQL tambahan yang mengharuskan Anda menjalankan perintah CREATE ENDPOINT.
- Amazon RDS memiliki dukungan terbatas untuk server tertaut. Saat mengonversi kode aplikasi SQL Server yang menggunakan server tertaut, AWS SCT mengkonversi kode aplikasi. Namun, pastikan untuk meninjau perilaku objek yang menggunakan server tertaut sebelum Anda menjalankan kode yang dikonversi.
- Selalu aktif digunakan.
- The AWS SCT laporan penilaian menyediakan metrik server untuk konversi. Metrik tentang instance SQL Server ini mencakup yang berikut:
 - Pencerminan data digunakan.
 - SQL Server Log Shipping dikonfigurasi.
 - Failover cluster digunakan.
 - Database Mail dikonfigurasi.
 - Layanan Pencarian Teks Lengkap digunakan. Amazon RDS for SQL Server memiliki pencarian teks lengkap terbatas, dan tidak mendukung pencarian semantik.
 - Layanan Kualitas Data (DQS) diinstal. Amazon RDS tidak mendukung DQS jadi kami sarankan Anda menginstal SQL Server pada instans Amazon EC2.

Hak istimewa untuk RDS untuk SQL Server sebagai target

Untuk bermigrasi ke RDS untuk SQL Server, buat pengguna database dan kemudian berikan hak istimewa yang diperlukan untuk setiap database. Anda dapat menggunakan contoh kode berikut ini.

```
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;
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *db_nama* dengan nama database target ini. Akhirnya, ganti *kata sandi_Anda* dengan kata sandi yang aman.

Sumber gudang data untuk AWS Schema Conversion Tool

AWS SCT dapat mengkonversi skema untuk gudang data sumber berikut ke target yang didukung. Untuk informasi tentang izin, koneksi, dan apa yang AWS SCT dapat dikonversi untuk digunakan dengan database target atau gudang data, lihat detail berikut ini.

Topik

- [Menggunakan Amazon Redshift sebagai sumber untuk AWS SCT](#)
- [Menggunakan Azure Synapse Analytics sebagai sumber AWS SCT](#)
- [Menggunakan BigQuery sebagai sumber untuk AWS SCT](#)
- [Menggunakan Greenplum Database sebagai sumber untuk AWS SCT](#)
- [Menggunakan Netezza sebagai sumber AWS SCT](#)
- [Menggunakan Oracle Data Warehouse sebagai sumber AWS SCT](#)
- [Menggunakan Snowflake sebagai sumber untuk AWS SCT](#)
- [Menggunakan Microsoft SQL Server Data Warehouse sebagai sumber untuk AWS SCT](#)
- [Menggunakan Teradata sebagai sumber AWS SCT](#)
- [Menggunakan Vertica sebagai sumber AWS SCT](#)

Menggunakan Amazon Redshift sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengoptimalkan kluster Amazon Redshift Anda. AWS SCT memberi Anda rekomendasi tentang pemilihan kunci distribusi dan sortir untuk kluster Amazon

Redshift Anda. Anda dapat mempertimbangkan proyek pengoptimalan Amazon Redshift sebagai AWS SCT proyek dengan sumber dan target menunjuk ke kluster Amazon Redshift yang berbeda.

Hak istimewa untuk Amazon Redshift sebagai database sumber

Hak istimewa berikut diperlukan untuk menggunakan Amazon Redshift sebagai sumber:

- PENGGUNAAN PADA SKEMA <schema_name>
- PILIH PADA SEMUA TABEL DALAM SKEMA <schema_name>
- PILIH PADA PG_CATALOG.PG_STATISTIC
- PILIH DI SVV_TABLE_INFO
- PILIH PADA TABEL STV_BLOCKLIST
- PILIH PADA TABEL STV_TBL_PERM
- PILIH DI SYS_SERVERLESS_USAGE
- PILIH PADA PG_DATABASE_INFO
- PILIH PADA PG_STATISTIC

Dalam contoh sebelumnya, ganti <schema_name>placeholder dengan nama skema sumber.

Untuk hak istimewa yang diperlukan untuk Amazon Redshift sebagai target, lihat. [Izin untuk Amazon Redshift sebagai target](#)

Menghubungkan ke Amazon Redshift sebagai sumber

Gunakan prosedur berikut untuk menyambung ke database sumber Amazon Redshift Anda dengan AWS Schema Conversion Tool

Untuk menyambung ke database sumber Amazon Redshift

1. DiAWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Amazon Redshift, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCTmenampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:

1. Untuk AWS Rahasia, pilih nama rahasianya.
2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi untuk database sumber Amazon Redshift, gunakan petunjuk berikut:

Parameter	Action
Nama server	Masukkan nama Domain Name System (DNS) atau alamat IP server database sumber Anda.
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis data	Masukkan nama database Amazon Redshift.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>

Parameter	Action
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none"> • Verifikasi sertifikat server: Pilih opsi ini untuk memverifikasi sertifikat server menggunakan toko kepercayaan. • Toko kepercayaan: Lokasi toko perwalian yang berisi sertifikat. Agar lokasi ini muncul di sini, pastikan untuk menambahkannya di Pengaturan global. <p>Untuk informasi selengkapnya tentang dukungan SSL untuk Amazon Redshift, lihat Mengonfigurasi opsi keamanan untuk koneksi.</p>
Simpan kata sandi	<p>AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.</p>
Jalur driver Redshift	<p>Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

5. Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
6. Pilih Hubungkan untuk terhubung ke database sumber Anda.

Pengaturan pengoptimalan Amazon Redshift

Untuk mengedit pengaturan pengoptimalan Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Amazon Redshift, lalu pilih Amazon Redshift — Amazon

Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk pengoptimalan Amazon Redshift.

Pengaturan pengoptimalan Amazon Redshift dalam AWS SCT menyertakan opsi untuk hal-hal berikut:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menetapkan jumlah maksimum tabel yang AWS SCT dapat diterapkan ke klaster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke klaster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node klaster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) di Panduan Manajemen Amazon Redshift.

AWS SCT mengubah semua tabel sumber Anda, bahkan jika jumlah tabel lebih dari yang dapat disimpan oleh klaster Amazon Redshift Anda. AWS SCT menyimpan kode dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka AWS SCT tampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke klaster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memilih strategi migrasi.

AWS merekomendasikan penggunaan cluster yang berbeda sebagai sumber dan target untuk proyek optimasi Anda. Sebelum memulai proses pengoptimalan Amazon Redshift, Anda membuat

salinan kluster Amazon Redshift sumber Anda. Anda dapat menyertakan data sumber Anda ke dalam salinan ini atau membuat cluster kosong.

Untuk strategi Migrasi, pilih Migrasi ke salinan untuk menyertakan data dari kluster sumber Anda di kluster target.

Untuk strategi Migrasi, pilih Migrasi ke papan tulis yang bersih untuk meninjau saran pengoptimalan. Setelah Anda menerima saran ini, migrasikan data sumber Anda ke kluster target.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci sortir dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk kolom KEY. Anda dapat memilih opsi ini hanya jika Anda memilih opsi Gunakan pengkodean kompresi.

- Untuk bekerja dengan optimasi tabel otomatis.

Optimalisasi tabel otomatis adalah proses penyetelan mandiri di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya menggunakan pengoptimalan tabel otomatis, pilih Strategi pengoptimalan di panel kiri. Kemudian pilih Use Amazon Redshift automatic table tuning, dan pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi pengoptimalan, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Menggunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilai (0-100). Nilai-nilai ini menentukan berat setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan memengaruhi pilihan kunci distribusi dan sortir. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Find small tables. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimalisasi konversi.

- Untuk Urutkan batas kolom kunci, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kejadian dari nilai yang paling umum untuk jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1—100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih statistik pengguna, pilih pengguna database yang statistik kueri Anda ingin menganalisis.

Menggunakan Azure Synapse Analytics sebagai sumber AWS SCT

Anda dapat menggunakannya AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Azure Synapse Analytics ke Amazon Redshift.

Hak istimewa untuk Azure Synapse Analytics sebagai database sumber

Hak istimewa berikut diperlukan untuk menggunakan gudang data Azure Synapse Analytics sebagai sumber:

- LIHAT DEFINISI
- LIHAT KEADAAN DATABASE

Terapkan hak istimewa untuk setiap database yang skemanya Anda konversi.

Menghubungkan ke Azure Synapse Analytics sebagai sumber

Gunakan prosedur berikut untuk terhubung ke gudang data Azure Synapse Analytics dengan file. AWS Schema Conversion Tool

Untuk menyambung ke gudang data Azure Synapse Analytics sebagai sumber

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Azure Synapse Analytics, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Rahasia, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi untuk gudang data Azure Synapse Analytics secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama server	Masukkan nama Domain Name Service (DNS) atau alamat IP server database sumber Anda.
SQL kolam renang	Masukkan nama pool Azure SQL.
Nama pengguna dan Kata Sandi	Masukkan kredensi database untuk terhubung ke server database sumber Anda. AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih

Parameter	Action
	<p>untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none"> • Sertifikat server kepercayaan: Pilih opsi ini untuk mempercayai sertifikat server. • Toko kepercayaan: Toko kepercayaan yang Anda siapkan di pengaturan Global.
Simpan kata sandi	<p>AWS SCTmenciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa memasukkan kata sandi.</p>

5. Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
6. Pilih Hubungkan untuk terhubung ke database sumber Anda.

Pengaturan konversi Azure Synapse Analytics ke Amazon Redshift

Untuk mengedit pengaturan konversi Azure Synapse Analytics ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Azure Synapse, lalu pilih Azure Synapse - Amazon Redshift. AWS SCTmenampilkan semua pengaturan yang tersedia untuk konversi Azure Synapse Analytics ke Amazon Redshift.

Pengaturan konversi Azure Synapse Analytics ke Amazon Redshift dalam opsi AWS SCT sertakan untuk hal berikut:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menetapkan jumlah maksimum tabel yang AWS SCT dapat diterapkan ke klaster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke klaster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node klaster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) di Panduan Manajemen Amazon Redshift.

AWS SCT mengubah semua tabel sumber Anda, bahkan jika ini lebih dari yang dapat disimpan oleh klaster Amazon Redshift Anda. AWS SCT menyimpan kode dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka AWS SCT tampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke klaster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasi partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan tampilan UNION ALL dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk satu tabel sumber.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasi setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang mencakup data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke klaster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah

maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci sortir dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk kolom KEY. Anda dapat memilih opsi ini hanya jika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Azure Synapse Analytics ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Azure Synapse Analytics ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Azure Synapse, lalu pilih Azure Synapse - Amazon Redshift. Di panel kiri, pilih Strategi pengoptimalan. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Azure Synapse Analytics ke Amazon Redshift.

Pengaturan pengoptimalan konversi Azure Synapse Analytics to Amazon Redshift AWS SCT termasuk opsi berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses penyetelan mandiri di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi pengoptimalan, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Menggunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilai (0-100). Nilai-nilai ini menentukan berat setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan memengaruhi pilihan kunci distribusi dan sortir. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Find small tables. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimalisasi konversi.

- Untuk Urutkan batas kolom kunci, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai condong lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kejadian dari nilai yang paling umum untuk jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1—100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih statistik pengguna, pilih pengguna database yang Anda ingin menganalisis statistik query.

Selain itu, pada tab Strategi pengoptimalan, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menggunakan BigQuery sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari BigQuery ke Amazon Redshift.

Hak istimewa untuk BigQuery sebagai sumber

Untuk menggunakan gudang BigQuery data sebagai sumber AWS SCT, buat akun layanan. Di Google Cloud, aplikasi menggunakan akun layanan untuk melakukan panggilan API resmi. Akun layanan berbeda dari akun pengguna. Untuk informasi selengkapnya, lihat [Akun layanan](#) di dokumentasi Google Cloud Identity dan Access Management.

Pastikan Anda memberikan peran berikut ke akun layanan Anda:

- BigQuery Admin
- Storage Admin

BigQuery Admin Peran ini memberikan izin untuk mengelola semua sumber daya dalam proyek. AWS SCT menggunakan peran ini untuk memuat BigQuery metadata Anda di proyek migrasi.

Storage Admin Peran memberikan kontrol penuh atas objek data dan bucket. Anda dapat menemukan peran ini di bawah **Cloud Storage**. AWS SCT menggunakan peran ini untuk mengekstrak data Anda BigQuery dan kemudian memuatnya ke Amazon Redshift.

Membuat file kunci akun layanan

1. Masuk ke konsol manajemen Google Cloud di <https://console.cloud.google.com/>.
2. Pada halaman [BigQuery API](#), pilih Aktifkan. Lewati langkah ini jika Anda melihat API Diaktifkan.
3. Pada halaman [Akun layanan](#), pilih proyek Anda, lalu pilih Buat akun layanan.
4. Pada halaman Detail akun layanan, masukkan nilai deskriptif untuk nama akun Layanan. Pilih Buat dan lanjutkan. Hibah akses akun layanan ini ke halaman proyek terbuka.
5. Untuk Pilih peran, pilih BigQuery, lalu pilih BigQueryAdmin.
6. Pilih Tambahkan peran lain. Untuk Pilih peran, pilih Cloud Storage, lalu pilih Storage Admin.
7. Pilih Lanjutkan, lalu pilih Selesai.
8. Pada halaman [Akun layanan](#), pilih akun layanan yang Anda buat.
9. Pilih Kunci, lalu pilih Buat kunci baru untuk Tambahkan kunci.

10. Pilih JSON, lalu pilih Buat. Pilih folder untuk menyimpan kunci pribadi Anda atau pilih folder default untuk diunduh di browser Anda.

Untuk mengekstrak data dari gudang BigQuery data, AWS SCT gunakan folder bucket Google Cloud Storage. Buat bucket ini sebelum memulai migrasi data. Masukkan path ke folder bucket Google Cloud Storage di kotak dialog Create Local task. Untuk informasi selengkapnya, lihat [Membuat, menjalankan, dan memantau AWS SCT tugas](#).

Menghubungkan ke BigQuery sebagai sumber

Gunakan prosedur berikut untuk terhubung ke BigQuery proyek sumber Anda dengan AWS Schema Conversion Tool.

Untuk menyambung ke gudang data BigQuery sumber

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih BigQuery, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk BigQuery proyek Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Untuk Jalur kunci, masukkan jalur ke file kunci akun layanan. Untuk informasi selengkapnya tentang membuat file ini, lihat [Hak istimewa untuk BigQuery sebagai sumber](#).
5. Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke BigQuery proyek sumber Anda.
6. Pilih Hubungkan untuk terhubung ke BigQuery proyek sumber Anda.

Keterbatasan penggunaan BigQuery sebagai sumber untuk AWS SCT

Keterbatasan berikut berlaku saat menggunakan BigQuery sebagai sumber untuk AWS SCT:

- AWS SCT tidak mendukung konversi subqueries dalam fungsi analitik.
- Anda tidak dapat menggunakan AWS SCT untuk mengkonversi BigQuery `SELECT AS STRUCT` dan `SELECT AS VALUE` pernyataan.
- AWS SCT tidak mendukung konversi dari jenis fungsi berikut:
 - Perkiraan agregat

- Bit
 - Debugging
 - Kueri federasi
 - Geografi
 - Hash
 - Matematika
 - Net
 - Agregat statistik
 - UUID
- AWS SCT memberikan dukungan terbatas untuk konversi fungsi string.
 - AWS SCT tidak mendukung konversi UNNEST operator.
 - Anda tidak dapat mengonversi operasi gabungan yang berkorelasi di AWS SCT.
 - AWS SCT tidak mendukung konversi QUALIFY,, WINDOWLIMIT, dan OFFSET klausa.
 - Anda tidak dapat menggunakan AWS SCT untuk mengkonversi ekspresi tabel umum rekursif.
 - AWS SCT tidak mendukung konversi INSERT pernyataan dengan subqueries di dalam VALUES klausa.
 - AWS SCT tidak mendukung konversi UPDATE pernyataan untuk bidang bersarang dan catatan berulang.
 - Anda tidak dapat menggunakan AWS SCT untuk mengkonversi STRUCT dan tipe ARRAY data.

BigQuery ke pengaturan konversi Amazon Redshift

Untuk mengedit BigQuery ke pengaturan konversi Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Google BigQuery, lalu pilih Google BigQuery - Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia BigQuery untuk konversi Amazon Redshift.

BigQuery ke pengaturan konversi Amazon Redshift dalam opsi AWS SCT sertakan untuk berikut ini:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS

SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menetapkan jumlah maksimum tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk kluster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node kluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke kluster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) di Panduan Manajemen Amazon Redshift.

AWS SCT mengubah semua tabel sumber Anda, bahkan jika ini lebih dari yang dapat disimpan oleh kluster Amazon Redshift Anda. AWS SCT menyimpan kode dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota kluster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka AWS SCT tampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke kluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci sortir dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk kolom KEY. Anda dapat memilih opsi ini hanya jika Anda memilih opsi Gunakan pengkodean kompresi.

BigQuery ke pengaturan pengoptimalan konversi Amazon Redshift

Untuk mengedit BigQuery ke pengaturan pengoptimalan konversi Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Google BigQuery, lalu pilih Google BigQuery - Amazon Redshift. Di panel kiri, pilih Strategi pengoptimalan. AWS SCT menampilkan pengaturan pengoptimalan konversi BigQuery untuk konversi Amazon Redshift.

BigQuery ke pengaturan pengoptimalan konversi Amazon Redshift dalam opsi AWS SCT sertakan untuk yang berikut ini:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses penyetelan mandiri di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi pengoptimalan, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Menggunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilai (0-100). Nilai-nilai ini menentukan berat setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan memengaruhi pilihan kunci distribusi dan sortir. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Find small tables. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimalisasi konversi.

- Untuk Urutkan batas kolom kunci, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai condong lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kejadian dari nilai yang paling umum untuk jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1—100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih statistik pengguna, pilih pengguna database yang Anda ingin menganalisis statistik query.

Selain itu, pada tab Strategi pengoptimalan, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menggunakan Greenplum Database sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Greenplum Database ke Amazon Redshift.

Hak istimewa untuk Greenplum Database sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan Greenplum Database sebagai sumber:

- TERHUBUNG DI DATABASE `<database_name>`
- PENGGUNAAN PADA SKEMA `<schema_name>`
- PILIH PADA `<schema_name>`. `<table_name>`
- PILIH PADA URUTAN `<schema_name>`. `<sequence_name>`

Pada contoh sebelumnya, ganti placeholder sebagai berikut:

- Ganti `database_name` dengan `nama` database sumber.

- Ganti *schema_name* dengan nama skema sumber.
- Ganti *table_name* dengan nama tabel sumber.
- Ganti *sequence_name* dengan nama nama urutan.

Menghubungkan ke Greenplum Database sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Greenplum Anda dengan AWS SCT.

Untuk menyambung ke database sumber Greenplum

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih SAP ASE, lalu pilih Next.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Rahasia, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan kredensi database sumber Greenplum secara manual, gunakan instruksi berikut:

Parameter	Action
Nama server	Masukkan nama Domain Name System (DNS) atau alamat IP server database sumber Anda.
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis data	Masukkan nama database Greenplum.

Parameter	Action
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none">• Verifikasi sertifikat server: Pilih opsi ini untuk memverifikasi sertifikat server menggunakan toko kepercayaan.• Toko kepercayaan: Lokasi toko perwalian yang berisi sertifikat.
Simpan kata sandi	<p>AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.</p>
Jalur driver Greenplum Database	<p>Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

5. Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
6. Pilih Hubungkan untuk terhubung ke database sumber Anda.

Pengaturan konversi Greenplum ke Amazon Redshift

Untuk mengedit pengaturan konversi Greenplum ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Greenplum, lalu pilih Greenplum - Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Greenplum ke Amazon Redshift.

Pengaturan konversi Greenplum ke Amazon Redshift AWS SCT termasuk opsi berikut:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menetapkan jumlah maksimum tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift target Anda.

Untuk jumlah maksimum tabel untuk kluster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node kluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke kluster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) di Panduan Manajemen Amazon Redshift.

AWS SCT mengubah semua tabel sumber Anda, bahkan jika ini lebih dari yang dapat disimpan oleh kluster Amazon Redshift Anda. AWS SCT menyimpan kode dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota kluster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka AWS SCT tampilkan pesan

peringatan. Selain itu, AWS SCT terapkan tabel ke klaster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasi partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan tampilan UNION ALL dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk satu tabel sumber.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasi setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang mencakup data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke klaster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci sortir dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk kolom KEY. Anda dapat memilih opsi ini hanya jika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Greenplum ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Greenplum ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Greenplum, lalu pilih Greenplum - Amazon Redshift. Di panel kiri, pilih Strategi pengoptimalan. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Greenplum ke Amazon Redshift.

Pengaturan pengoptimalan konversi Greenplum ke Amazon Redshift dalam AWS SCT menyertakan opsi untuk berikut ini:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses penyetelan mandiri di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi pengoptimalan, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Menggunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilai (0-100). Nilai-nilai ini menentukan berat setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan memengaruhi pilihan kunci distribusi dan sortir. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Find small tables. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimalisasi konversi.

- Untuk Urutkan batas kolom kunci, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai condong lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kejadian dari nilai yang paling umum untuk jumlah total catatan.

- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1—100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih statistik pengguna, pilih pengguna database yang Anda ingin menganalisis statistik query.

Selain itu, pada tab Strategi pengoptimalan, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menggunakan Netezza sebagai sumber AWS SCT

Anda dapat menggunakannya AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Netezza ke Amazon Redshift.

Hak istimewa untuk Netezza sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan Netezza sebagai sumber:

- pilih pada tampilan `system.definition_schema.system`
- pilih pada tabel `system.definition_schema.system`
- pilih pada tabel `system.definition_schema.management`
- daftar di `<database_name>`
- daftar di `<schema_name>`
- daftar di `<database_name>.all.table`
- daftar di `<database_name>tabel.all.external`
- daftar di `<database_name>.all.view`
- daftar pada tampilan `<database_name>.all.materialized`
- daftar `<database_name>di.all.procedure`
- daftar di `<database_name>.all.sequence`
- daftar `<database_name>di.all.function`
- daftar di `<database_name>.all.aggregate`

Pada contoh sebelumnya, ganti placeholder sebagai berikut:

- Ganti *database_name* dengan *nama* database sumber.
- Ganti *schema_name* dengan nama skema sumber.

AWS SCT membutuhkan akses ke tabel dan tampilan sistem berikut. Anda dapat memberikan akses ke benda-benda ini alih-alih memberikan akses ke `system.definition_schema.system view` dan `system.definition_schema.system tables` dalam daftar sebelumnya.

- pilih pada `system.definition_schema._t_agregat`
- pilih pada `system.definition_schema._kelas`
- pilih pada `system.definition_schema._t_constraint`
- pilih pada `system.definition_schema._const_relattr`
- pilih pada `system.definition_schema._basisdata`
- pilih pada `system.definition_schema._grpobj_priv`
- pilih pada `system.definition_schema._grpusr`
- pilih pada `system.definition_schema._hist_config`
- pilih pada `system.definition_schema._t_object`
- pilih pada `system.definition_schema._t_object_classes`
- pilih pada `system.definition_schema._t_proc`
- pilih pada `system.definition_schema._t_type`
- pilih pada `system.definition_schema._t_user`
- pilih pada `system.definition_schema._usrobj_priv`
- pilih pada `system.definition_schema._vt_urutan`
- pilih pada `system.definition_schema._v_agregat`
- pilih pada `system.definition_schema._v_constraint_tergantung`
- pilih pada `system.definition_schema._v_database`
- pilih pada `system.definition_schema._v_tipe data`
- pilih pada `system.definition_schema._v_dslice`
- pilih pada `system.definition_schema._v_fungsi`
- pilih pada `system.definition_schema._v_grup`
- pilih pada `system.definition_schema._v_obj_relation`

- pilih pada `system.definition_schema. _v_obj_relation_xdb`
- pilih pada `system.definition_schema. _v_prosedur`
- pilih pada `system.definition_schema. _v_relation_column`
- pilih pada `system.definition_schema. _v_hubungan_keydata`
- pilih pada `system.definition_schema. _v_relobjclasses`
- pilih pada `system.definition_schema. _v_schema_xdb`
- pilih pada `system.definition_schema. _v_urutan`
- pilih pada `system.definition_schema. _v_sinonim`
- pilih pada `system.definition_schema. _v_system_info`
- pilih pada `system.definition_schema. _v_sys_constraint`
- pilih pada `system.definition_schema. _v_sys_object_dslice_info`
- pilih pada `system.definition_schema. _v_sys_user`
- pilih pada `system.definition_schema. _v_tabel`
- pilih pada `system.definition_schema. _v_table_constraint`
- pilih pada `system.definition_schema. _v_table_dist_map`
- pilih pada `system.definition_schema. _v_table_organize_column`
- pilih pada `system.definition_schema. _v_table_storage_stat`
- pilih pada `system.definition_schema. _v_pengguna`
- pilih pada `system.definition_schema. _v_lihat`
- pilih pada `system.information_schema. _v_relation_column`
- pilih pada `system.information_schema. _v_tabel`
- pilih pada `$hist_column_access_*`

Menghubungkan ke Netezza sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Netezza Anda dengan AWS Schema Conversion Tool

Untuk menyambung ke database sumber Netezza

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Netezza, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Rahasia, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi database sumber Netezza secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama server	Masukkan nama Domain Name System (DNS) atau alamat IP server database sumber Anda.
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Simpan kata sandi	AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan

Parameter	Action
	mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.
Jalur pengemudi Netezza	<p>Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

- Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
- Pilih Hubungkan untuk terhubung ke database sumber Anda.

Mengonfigurasi replikasi data yang sedang berlangsung

Setelah mengonversi skema database Netezza dan menerapkannya ke database Amazon Redshift, Anda dapat memigrasi data dengan agen ekstraksi data. AWS SCT Agen mengekstrak data Anda dan mengunggahnya ke bucket Amazon S3 Anda. Anda kemudian dapat menggunakan AWS SCT untuk menyalin data dari Amazon S3 ke Amazon Redshift.

Jika data dalam database sumber berubah selama proses migrasi, Anda dapat menangkap perubahan yang sedang berlangsung dengan agen ekstraksi AWS SCT data Anda. Kemudian Anda dapat mereplikasi perubahan yang sedang berlangsung ini di database target Anda setelah Anda menyelesaikan migrasi data awal. Proses ini disebut replikasi data yang sedang berlangsung atau mengubah data capture (CDC).

Untuk mengonfigurasi replikasi data yang sedang berlangsung untuk migrasi dari Netezza ke Amazon Redshift

- Di database sumber Anda, buat database riwayat. Anda dapat menggunakan contoh kode berikut di antarmuka baris perintah Netezza (CLI).

```
nzhistcreatedb -d history_database_name -t query -v 1 -u load_user -o histdb_owner
-p your_password
```

Pada contoh sebelumnya, ganti *history_database_name* dengan nama database histori Anda. Selanjutnya, ganti *load_user* dengan nama pengguna yang telah Anda tetapkan untuk memuat data riwayat ke database. Kemudian, ganti *histdb_owner* dengan nama pengguna yang telah Anda definisikan sebagai pemilik database history. Pastikan Anda telah membuat pengguna ini dan memberikan CREATE DATABASE izin. Terakhir, ganti kata *sandi Anda* dengan kata *sandi* yang aman.

2. Konfigurasi pencatatan riwayat. Untuk melakukannya, gunakan contoh kode berikut.

```
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;
```

Pada contoh sebelumnya, ganti *history_configuration_name* dan *history_database_name* dengan nama konfigurasi histori dan database histori Anda. Selanjutnya, ganti *load_user* dengan nama pengguna yang telah Anda tetapkan untuk memuat data riwayat ke database. Kemudian, ganti kata *sandi Anda* dengan kata *sandi* yang aman.

3. Berikan izin baca untuk semua tabel dalam database riwayat. Anda dapat menggunakan contoh kode berikut untuk memberikan SELECT izin.

```
GRANT SELECT ON history_database_name.ALL.TABLE TO your_user;
```

Pada contoh sebelumnya, ganti *history_database_name* dengan nama database histori Anda. Selanjutnya, ganti *your_user* dengan nama pengguna dengan izin minimal untuk bekerja dengan database Netezza Anda. Anda menggunakan kredensi pengguna database ini di AWS SCT

4. Kumpulkan statistik untuk setiap tabel dalam skema sumber Anda untuk mendapatkan informasi tentang kardinalitas kolom. Anda dapat menggunakan perintah berikut untuk menghasilkan statistik dalam database sejarah Anda.

```
GENERATE STATISTICS on "schema_name".table_name";
```

Pada contoh sebelumnya, ganti *schema_name* dan *table_name* dengan nama skema dan tabel database Anda.

5. Pastikan Anda menyelesaikan prasyarat dengan menjalankan kueri berikut:

```
SELECT COUNT(*)  
FROM history_database_name.history_schema_name."$hist_column_access_N";
```

Pada contoh sebelumnya, ganti *history_database_name* dan *history_schema_name* dengan nama database riwayat dan skema Anda. Selanjutnya, ganti *N* dengan nomor versi database riwayat Anda. Untuk informasi selengkapnya tentang versi database histori, lihat Dokumentasi [IBM Netezza](#).

6. Instal agen ekstraksi data Anda. Untuk informasi selengkapnya, lihat [Instalasi agen ekstraksi](#).

Pastikan {*working.folder*} parameter dalam *settings.properties* file untuk semua instance ekstraktor menunjuk ke folder yang sama. Dalam hal ini, ekstraktor Anda dapat mengoordinasikan sesi CDC dan menggunakan titik transaksi tunggal untuk semua subtugas.

7. Daftarkan agen ekstraksi data Anda. Untuk informasi selengkapnya, lihat [Mendaftarkan agen ekstraksi dengan AWS Schema Conversion Tool](#).

8. Buat tugas CDC Anda. Untuk informasi selengkapnya, lihat [Membuat, menjalankan, dan memantau AWS SCT tugas](#).

- a. Buka proyek Anda di AWS SCT. Di panel kiri, pilih tabel sumber Anda. Buka menu konteks (klik kanan), dan pilih Buat tugas lokal.

- b. Untuk Nama tugas, masukkan nama deskriptif untuk tugas migrasi data Anda.

- c. Untuk mode Migrasi, pilih Ekstrak, unggah, dan salin.

- d. Pilih Aktifkan CDC.

- e. Pilih tab pengaturan CDC dan tentukan ruang lingkup dan jadwal sesi CDC.

- f. Pilih Test task untuk memverifikasi bahwa Anda dapat terhubung ke folder kerja, bucket Amazon S3, dan gudang data Amazon Redshift.

- g. Pilih Buat untuk membuat tugas Anda.

- h. Pilih tab Tugas, pilih tugas Anda dari daftar, dan pilih Mulai.

9. AWS SCT tugas mempertahankan konsistensi transaksional pada database target. Agen ekstraksi data mereplikasi transaksi dari sumber dalam urutan ID transaksi.

Jika Anda menghentikan salah satu sesi migrasi atau jika gagal, maka pemrosesan CDC juga berhenti.

Pengaturan konversi Netezza ke Amazon Redshift

Untuk mengedit pengaturan konversi Netezza ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Netezza, lalu pilih Netezza - Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Netezza ke Amazon Redshift.

Pengaturan konversi Netezza ke Amazon Redshift dalam opsi AWS SCT sertakan untuk yang berikut ini:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menetapkan jumlah maksimum tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk kluster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node kluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke kluster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) di Panduan Manajemen Amazon Redshift.

AWS SCT mengubah semua tabel sumber Anda, bahkan jika ini lebih dari yang dapat disimpan oleh kluster Amazon Redshift Anda. AWS SCT menyimpan kode dikonversi dalam proyek Anda

dan tidak menerapkannya ke database target. Jika Anda mencapai kuota kluster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka AWS SCT tampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke kluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci sortir dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk kolom KEY. Anda dapat memilih opsi ini hanya jika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Netezza to Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Netezza ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Netezza, lalu pilih Netezza - Amazon Redshift. Di panel kiri, pilih Strategi pengoptimalan. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Netezza ke Amazon Redshift.

Pengaturan pengoptimalan konversi Netezza to Amazon Redshift dalam opsi AWS SCT sertakan untuk yang berikut ini:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses penyetelan mandiri di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi pengoptimalan, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Menggunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilai (0-100). Nilai-nilai ini menentukan berat setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan memengaruhi pilihan kunci distribusi dan sortir. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Find small tables. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimalisasi konversi.

- Untuk Urutkan batas kolom kunci, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai condong lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kejadian dari nilai yang paling umum untuk jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1—100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih statistik pengguna, pilih pengguna database yang Anda ingin menganalisis statistik query.

Selain itu, pada tab Strategi pengoptimalan, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menggunakan Oracle Data Warehouse sebagai sumber AWS SCT

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Oracle Data Warehouse ke Amazon Redshift atau Amazon Redshift dan AWS Glue digunakan dalam kombinasi.

Hak istimewa untuk Oracle Data Warehouse sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan Oracle Data Warehouse sebagai sumber:

- menghubungkan
- `select_catalog_role`
- pilih kamus apa saja

Menghubungkan ke Oracle Data Warehouse sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber gudang data Oracle Anda dengan file. AWS Schema Conversion Tool

Untuk menyambung ke database sumber Oracle Data Warehouse

1. DiAWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Oracle, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWSRahasia, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi gudang data sumber Oracle secara manual, gunakan instruksi berikut:

Parameter	Action
Jenis	<p>Pilih jenis koneksi ke database Anda. Tergantung pada jenis Anda, berikan informasi tambahan berikut:</p> <ul style="list-style-type: none">• SID<ul style="list-style-type: none">• Nama server: Nama Domain Name System (DNS) atau alamat IP server database sumber Anda.• Port server: Port yang digunakan untuk terhubung ke server database sumber Anda.• Oracle SID: Oracle Sistem ID (SID). Untuk menemukan Oracle SID, kirimkan query berikut ke database Oracle Anda: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre>• Nama Layanan<ul style="list-style-type: none">• Nama server: Nama DNS atau alamat IP server database sumber Anda.• Port server: Port yang digunakan untuk terhubung ke server database sumber Anda.• Nama Layanan: Nama layanan Oracle untuk terhubung• Alias TNS<ul style="list-style-type: none">• Jalur file TNS: Jalur ke file yang berisi informasi koneksi nama Transparan Jaringan Substrat (TNS).• Jalur berkas TNS: Alias TNS dari file ini untuk digunakan untuk terhubung ke database sumber.• TNS menghubungkan identifier<ul style="list-style-type: none">• TNS connect identifier: Pengenal untuk informasi koneksi TNS terdaftar.

Parameter	Action
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none">• Otentikasi SSL: Pilih opsi ini untuk menggunakan otentikasi SSL untuk koneksi.• Toko kepercayaan: Lokasi toko perwalian yang berisi sertifikat.• Toko kunci: Lokasi toko kunci yang berisi kunci pribadi dan sertifikat. Nilai ini diperlukan jika otentikasi SSL dipilih dan sebaliknya opsional.
Simpan kata sandi	<p>AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.</p>

Parameter	Action
Jalur pengemudi Oracle	<p>Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

- Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
- Pilih Hubungkan untuk terhubung ke database sumber Anda.

Pengaturan konversi Oracle Data Warehouse ke Amazon Redshift

Untuk mengedit pengaturan konversi Oracle Data Warehouse ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Oracle, lalu pilih Oracle - Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Oracle Data Warehouse ke Amazon Redshift.

Pengaturan konversi Oracle Data Warehouse ke Amazon Redshift dalam opsi AWS SCT sertakan untuk yang berikut ini:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menetapkan jumlah maksimum tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke klaster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node klaster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) di Panduan Manajemen Amazon Redshift.

AWS SCT mengubah semua tabel sumber Anda, bahkan jika ini lebih dari yang dapat disimpan oleh klaster Amazon Redshift Anda. AWS SCT menyimpan kode dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka AWS SCT tampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke klaster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasi partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan tampilan UNION ALL dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk satu tabel sumber.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasi setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang mencakup data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke klaster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk mengonversi fungsi pemformatan tipe data seperti TO_CHAR, TO_DATE, dan TO_NUMBER dengan elemen format datetime yang tidak didukung Amazon Redshift. Secara default, AWS SCT gunakan fungsi paket ekstensi untuk meniru penggunaan elemen format yang tidak didukung ini dalam kode yang dikonversi.

Model format datetime di Oracle menyertakan lebih banyak elemen dibandingkan dengan string format datetime di Amazon Redshift. Jika kode sumber hanya menyertakan elemen format datetime yang didukung Amazon Redshift, Anda tidak memerlukan fungsi paket ekstensi dalam

kode yang dikonversi. Untuk menghindari penggunaan fungsi paket ekstensi dalam kode yang dikonversi, pilih Elemen format tipe data yang Anda gunakan dalam kode Oracle mirip dengan string format datetime di Amazon Redshift. Dalam hal ini, kode yang dikonversi bekerja lebih cepat.

Model format numerik di Oracle menyertakan lebih banyak elemen dibandingkan dengan string format numerik di Amazon Redshift. Jika kode sumber hanya menyertakan elemen format numerik yang didukung Amazon Redshift, Anda tidak memerlukan fungsi paket ekstensi dalam kode yang dikonversi. Untuk menghindari penggunaan fungsi paket ekstensi dalam kode yang dikonversi, pilih Elemen format numerik yang Anda gunakan dalam kode Oracle mirip dengan string format numerik di Amazon Redshift. Dalam hal ini, kode yang dikonversi bekerja lebih cepat.

- Untuk mengkonversi Oracle LEAD dan fungsi LAG analitik. Secara default, AWS SCT memunculkan item tindakan untuk masing-masing LEAD dan LAG fungsi.

Ketika kode sumber Anda tidak menggunakan nilai default untuk offset dalam fungsi-fungsi ini, AWS SCT dapat meniru penggunaan fungsi-fungsi ini dengan NVL fungsi. Untuk melakukannya, pilih Gunakan fungsi NVL untuk meniru perilaku fungsi Oracle LEAD dan LAG.

- Untuk meniru perilaku kunci primer dan unik di kluster Amazon Redshift Anda, pilih Emulate perilaku kunci primer dan unik.

Amazon Redshift tidak menerapkan kunci unik dan utama serta menggunakannya untuk tujuan informasi saja. Jika Anda menggunakan kendala ini dalam kode Anda, maka pastikan bahwa AWS SCT mengemulasi perilaku mereka dalam kode dikonversi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci sortir dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk kolom KEY. Anda dapat memilih opsi ini hanya jika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Oracle Data Warehouse ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Oracle Data Warehouse ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Oracle, lalu pilih Oracle - Amazon Redshift. Di panel kiri, pilih Strategi pengoptimalan. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Oracle Data Warehouse ke Amazon Redshift.

Pengaturan pengoptimalan konversi Oracle Data Warehouse ke Amazon Redshift dalam opsi AWS SCT sertakan untuk yang berikut ini:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses penyetelan mandiri di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi pengoptimalan, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Menggunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilai (0-100). Nilai-nilai ini menentukan berat setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan memengaruhi pilihan kunci distribusi dan sortir. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Find small tables. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimalisasi konversi.

- Untuk Urutkan batas kolom kunci, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai condong lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kejadian dari nilai yang paling umum untuk jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1—100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih statistik pengguna, pilih pengguna database yang Anda ingin menganalisis statistik query.

Selain itu, pada tab Strategi pengoptimalan, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menggunakan Snowflake sebagai sumber untuk AWS SCT

Anda dapat menggunakannya AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Snowflake ke Amazon Redshift.

Hak istimewa untuk Snowflake sebagai database sumber

Anda dapat membuat peran dengan hak istimewa dan memberikan peran ini nama pengguna dengan menggunakan SECURITYADMIN peran dan konteks SECURITYADMIN sesi.

Contoh berikut menciptakan hak istimewa minimal dan memberikannya kepada pengguna.

`min_privs`

```
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;
```

Pada contoh sebelumnya, ganti placeholder sebagai berikut:

- Ganti *role_name* dengan nama peran dengan hak istimewa hanya-baca.
- Ganti *db_name* dengan nama database sumber.
- Ganti *schema_name* dengan nama skema sumber.
- Ganti *datawarehouse_name* dengan nama gudang data yang diperlukan.
- Ganti *min_privs* dengan nama pengguna yang memiliki hak istimewa minimal.

The DEFAULT_ROLE dan DEFAULT_WAREHOUSE parameter yang kunci-sensitif.

Mengonfigurasi akses aman ke Amazon S3

Kebijakan manajemen keamanan dan akses untuk bucket Amazon S3 memungkinkan Snowflake mengakses, membaca data dari, dan menulis data ke bucket S3. Anda dapat mengonfigurasi akses aman ke bucket Amazon S3 pribadi menggunakan jenis objek Snowflake STORAGE INTEGRATION. Objek integrasi penyimpanan Snowflake mendelegasikan tanggung jawab otentikasi ke identitas Snowflake dan entitas manajemen akses.

Untuk informasi selengkapnya, lihat [Mengonfigurasi Integrasi Penyimpanan Snowflake untuk Mengakses Amazon S3 dalam dokumentasi Snowflake](#).

Menghubungkan ke kepingan salju sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Anda dengan fileAWS Schema Conversion Tool.

Untuk menyambung ke database sumber Snowflake

1. DiAWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Snowflake, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCTmenampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWSRahasia, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi gudang data sumber Snowflake secara manual, gunakan instruksi berikut:

Parameter	Action
Nama server	Masukkan nama Domain Name System (DNS) atau alamat IP server database sumber Anda.
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis data	Masukkan nama database Snowflake.
Nama pengguna dan Kata Sandi	Masukkan kredensi database untuk terhubung ke server database sumber Anda.

Parameter	Action
	AWS SCT menyimpan kata sandi Anda dalam format terenkripsi hanya jika Anda secara eksplisit memintanya.
Gunakan SSL	<p>Pilih opsi ini jika Anda ingin menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none"> • Jalur kunci pribadi: Lokasi kunci pribadi. • Frasa sandi: Frasa sandi untuk kunci pribadi. <p>Untuk informasi selengkapnya tentang dukungan SSL untuk Snowflake, lihat Mengonfigurasi opsi keamanan untuk koneksi.</p>
Simpan kata sandi	AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Jika Anda mengatur opsi ini, Anda dapat menyimpan kata sandi database. Melakukan ini berarti Anda dapat terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.
Jalur pengemudi kepingan salju	<p>Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

5. Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
6. Pilih Hubungkan untuk terhubung ke database sumber Anda.

Keterbatasan untuk Snowflake sebagai sumber

Berikut ini adalah batasan saat menggunakan Snowflake sebagai sumber untuk: AWS SCT

- Pengidentifikasi objek harus unik dalam konteks jenis objek dan objek induk:

Basis Data

Pengenal skema harus unik dalam database.

Skema

Objek pengidentifikasi seperti untuk tabel dan tampilan harus unik dalam skema.

Tabel/Tampilan

Pengidentifikasi kolom harus unik dalam tabel.

- Jumlah maksimum tabel untuk tipe node cluster besar dan xlarge adalah 9.900. Untuk tipe node cluster 8xlarge, jumlah maksimum tabel adalah 100.000. Batas tersebut mencakup tabel sementara, baik yang ditentukan pengguna maupun dibuat oleh Amazon Redshift selama pemrosesan kueri atau pemeliharaan sistem. Untuk informasi selengkapnya, lihat [kuota Amazon Redshift](#) di Panduan Pengelolaan Klaster Amazon Redshift.
- Untuk prosedur yang disimpan, jumlah maksimum argumen input dan output adalah 32.

Jenis data sumber untuk Snowflake

Setelah itu, Anda dapat menemukan tipe data sumber Snowflake yang didukung saat menggunakan AWS SCT dan pemetaan default ke target Amazon Redshift.

Jenis data kepingan salju	Tipe data Amazon Redshift
NUMBER	NUMERIK (38)
NOMOR (p)	Jika p adalah =< 4, maka SMALLINT Jika p adalah => 5 dan =< 9, maka INTEGER Jika p adalah => 10 dan =< 18, maka BIGINT Jika p adalah => 19 maka NUMERIC (p)
NOMOR (p, 0)	Jika p adalah =< 4, maka SMALLINT

Jenis data kepingan salju	Tipe data Amazon Redshift
	<p>Jika p adalah => 5 dan =< 9, maka INTEGER</p> <p>Jika p adalah => 10 dan =< 18, maka BIGINT</p> <p>Jika p adalah => 19 maka: NUMERIC (p, 0)</p>
NOMOR (p, s)	<p>Jika p adalah => 1 dan =< 38, dan jika s adalah => 1 dan =< 37, maka</p> <p>NUMERIK (p, s)</p>
MENGAMBANG	MENGAMBANG
<p>TEXT</p> <p>Karakter Unicode hingga 16.777.216 byte; hingga 4 byte per karakter.</p>	VARCHAR (MAKS)
<p>TEKS (p)</p> <p>Karakter Unicode hingga 65.535 byte; hingga 4 byte per karakter.</p>	Jika p adalah =< 65,535 kemudian, VARCHAR (p)
<p>TEKS (p)</p> <p>Karakter Unicode hingga 16.777.216 byte; hingga 4 byte per karakter.</p>	Jika p adalah => 65,535 dan =< 16,777,216 kemudian, VARCHAR (MAX)
<p>BINER</p> <p>Karakter byte tunggal hingga 8.388.608 byte; 1 byte per karakter.</p>	VARCHAR (MAKS)
<p>BINER (p)</p> <p>Karakter byte tunggal hingga 65.535 byte; 1 byte per karakter.</p>	VARCHAR (p)

Jenis data kepingan salju	Tipe data Amazon Redshift
BINER (p) Karakter byte tunggal hingga 8.388.608 byte; 1 byte per karakter.	VARCHAR (MAKS)
BOOLEAN	BOOLEAN
TANGGAL	DATE
WAKTU Nilai waktu antara 00:00:00 dan 23:59:59.999 999999.	VARCHAR (18)
WAKTU (f) Nilai waktu antara 00:00:00 dan 23:59:59.9 (f).	VARCHAR (n) — 9+ dt-attr-1
TIMESTAMP_NTZ	TIMESTAMP
TIMESTAMP_TZ	STEMAMPTZ

Pengaturan konversi Snowflake ke Amazon Redshift

Untuk mengedit setelan konversi Snowflake ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Snowflake, lalu pilih Snowflake — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Snowflake ke Amazon Redshift.

Pengaturan konversi Snowflake ke Amazon Redshift dalam opsi AWS SCT sertakan untuk yang berikut ini:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS

SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menetapkan jumlah maksimum tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk kluster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node kluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke kluster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) di Panduan Manajemen Amazon Redshift.

AWS SCT mengubah semua tabel sumber Anda, bahkan jika ini lebih dari yang dapat disimpan oleh kluster Amazon Redshift Anda. AWS SCT menyimpan kode dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota kluster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka AWS SCT tampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke kluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci sortir dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk kolom KEY. Anda dapat memilih opsi ini hanya jika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Snowflake ke Amazon Redshift

Untuk mengedit setelan pengoptimalan konversi Snowflake ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Snowflake, lalu pilih Snowflake — Amazon Redshift. Di panel kiri, pilih Strategi pengoptimalan. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Snowflake ke Amazon Redshift.

Setelan pengoptimalan konversi Snowflake ke Amazon Redshift dalam opsi AWS SCT sertakan untuk yang berikut ini:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses penyetelan mandiri di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi pengoptimalan, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Menggunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilai (0-100). Nilai-nilai ini menentukan berat setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan memengaruhi pilihan kunci distribusi dan sortir. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Find small tables. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimalisasi konversi.

- Untuk Urutkan batas kolom kunci, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai condong lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kejadian dari nilai yang paling umum untuk jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1—100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih statistik pengguna, pilih pengguna database yang Anda ingin menganalisis statistik query.

Selain itu, pada tab Strategi pengoptimalan, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menggunakan Microsoft SQL Server Data Warehouse sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Microsoft SQL Server DW ke Amazon Redshift atau Amazon Redshift dan AWS Glue digunakan dalam kombinasi.

Hak istimewa untuk Microsoft SQL Server Data Warehouse sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan gudang data Microsoft SQL Server sebagai sumber:

- LIHAT DEFINISI
- LIHAT KEADAAN DATABASE
- PILIH PADA SKEMA:: <schema_name>

Pada contoh sebelumnya, ganti <source_schema>placeholder dengan nama source source_schema.

Ulangi hibah untuk setiap database yang skema Anda konversi.

Selain itu, berikan yang berikut, dan jalankan hibah pada database master:

- LIHAT STATUS SERVER

Batasan untuk SQL Server Data Warehouse sebagai sumber

Menggunakan Microsoft SQL Server Parallel Data Warehouse (PDW) sebagai sumber saat ini tidak didukung.

Menghubungkan ke SQL Server Data Warehouse sebagai sumber

Gunakan prosedur berikut untuk menyambung ke database sumber SQL Server Data Warehouse Anda dengan AWS Schema Conversion Tool.

Untuk menyambung ke database sumber SQL Server Data Warehouse

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Microsoft SQL Server, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Rahasia, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi sambungan gudang data sumber Microsoft SQL Server secara manual, gunakan petunjuk berikut ini:

Parameter	Action
Nama server	Masukkan nama Domain Name Service (DNS) atau alamat IP server database sumber Anda.
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Nama instans	Masukkan nama instans untuk gudang data SQL Server.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none">• Sertifikat server kepercayaan: Pilih opsi ini untuk mempercayai sertifikat server.• Toko kepercayaan: Toko kepercayaan yang Anda siapkan di pengaturan Global.
Simpan kata sandi	AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.

Parameter	Action
Jalur driver SQL Server	<p>Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

- Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
- Pilih Hubungkan untuk terhubung ke database sumber Anda.

Pengaturan konversi SQL Server Data Warehouse ke Amazon Redshift

Untuk mengedit pengaturan konversi SQL Server Data Warehouse ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Microsoft SQL Server, lalu pilih Microsoft SQL Server — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi SQL Server Data Warehouse ke Amazon Redshift.

Pengaturan konversi SQL Server Data Warehouse ke Amazon Redshift dalam opsi AWS SCT sertakan untuk yang berikut ini:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menetapkan jumlah maksimum tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke klaster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node klaster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) di Panduan Manajemen Amazon Redshift.

AWS SCT mengubah semua tabel sumber Anda, bahkan jika ini lebih dari yang dapat disimpan oleh klaster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka AWS SCT tampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke klaster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasi partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan tampilan UNION ALL dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk satu tabel sumber.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasi setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang mencakup data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke klaster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci sortir dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk kolom KEY. Anda dapat memilih opsi ini hanya jika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi SQL Server Data Warehouse ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi SQL Server Data Warehouse ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Microsoft SQL Server, lalu pilih Microsoft SQL Server — Amazon Redshift. Di panel kiri, pilih Strategi pengoptimalan. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi SQL Server Data Warehouse ke Amazon Redshift.

Pengaturan pengoptimalan konversi SQL Server Data Warehouse ke Amazon Redshift AWS SCT termasuk opsi untuk berikut ini:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses penyetelan mandiri di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi pengoptimalan, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Menggunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilai (0-100). Nilai-nilai ini menentukan berat setiap strategi.

Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan

memengaruhi pilihan kunci distribusi dan sortir. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Find small tables. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimalisasi konversi.

- Untuk Urutkan batas kolom kunci, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai condong lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kejadian dari nilai yang paling umum untuk jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1—100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih statistik pengguna, pilih pengguna database yang Anda ingin menganalisis statistik query.

Selain itu, pada tab Strategi pengoptimalan, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menggunakan Teradata sebagai sumber AWS SCT

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Teradata ke Amazon Redshift atau Amazon Redshift dan digunakan dalam kombinasi. AWS Glue

Hak istimewa untuk Teradata sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan Teradata sebagai sumber:

- PILIH DI DBC

- PILIH DI SYSUDTLIB
- PILIH PADA SYSLIB
- PILIH PADA <source_database>
- BUAT PROSEDUR PADA <source_database>

Pada contoh sebelumnya, ganti <source_database>placeholder dengan nama database sumber.

AWS SCTmembutuhkan CREATE PROCEDURE hak istimewa untuk melakukan PROSEDUR BANTUAN terhadap semua prosedur dalam database sumber. AWS SCTtidak menggunakan hak istimewa ini untuk membuat objek baru dalam database Teradata sumber Anda.

Menghubungkan ke Teradata sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Teradata Anda dengan. AWS Schema Conversion Tool

Untuk menyambung ke database sumber Teradata

1. DiAWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Teradata, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCTmenampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWSRahasia, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi database sumber Teradata secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama koneksi	Masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
Nama server	Masukkan nama Domain Name System (DNS) atau alamat IP server database sumber Anda.
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis data	Masukkan nama database Teradata.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Simpan kata sandi	AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.
Enkripsi data	Pilih opsi ini untuk mengenkripsi data yang Anda tukarkan dengan database. Jika Anda memilih opsi ini, maka nomor port 443 digunakan untuk mentransfer data terenkripsi antara AWS SCT dan database Teradata Anda.

Parameter	Action
Jalur pengemudi Teradata	<p>Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

- Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
- Pilih Hubungkan untuk terhubung ke database sumber Anda.

Menggunakan otentikasi LDAP dengan sumber Teradata

Untuk mengatur autentikasi Lightweight Directory Access Protocol (LDAP) untuk pengguna Teradata yang menjalankan Microsoft Active Directory di Windows, gunakan prosedur berikut.

Dalam prosedur berikut, domain Active Directory adalah `test.local.com`. Server WindowsDC, dan dikonfigurasi dengan pengaturan default. Skrip berikut membuat akun `test_ldap` Active Directory, dan akun ini menggunakan `test_ldap` kata sandi.

Untuk mengatur otentikasi LDAP untuk pengguna Teradata yang menjalankan Microsoft Active Directory pada Windows

- Di `/opt/teradata/tdat/tdgss/site` direktori, edit file `TdgssUserConfigFile.xml`. Ubah bagian LDAP menjadi yang berikut.

```
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"
```

- Terapkan perubahan dengan menjalankan konfigurasi sebagai berikut.

```
#cd /opt/teradata/tdgss/bin
#./run_tdgssconfig
```

3. Uji konfigurasi dengan menjalankan perintah berikut.

```
# /opt/teradata/tdat/tdgss/14.10.03.01/bin/tdsbind -u test_ldap -w test_ldap
```

Outputnya harus serupa dengan berikut ini.

```
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. Restart TPA menggunakan perintah berikut.

```
#tpareset -f "use updated TDGSSCONFIG GDO"
```

5. Buat pengguna yang sama dalam database Teradata seperti di Active Directory, seperti yang ditunjukkan berikut.

```
CREATE USER test_ldap AS PERM=1000, PASSWORD=test_ldap;
GRANT LOGON ON ALL TO test WITH NULL PASSWORD;
```

Jika Anda mengubah kata sandi pengguna di Active Directory untuk pengguna LDAP Anda, tentukan kata sandi baru ini selama koneksi ke Teradata dalam mode LDAP. Dalam mode DEFAULT, Anda terhubung ke Teradata dengan menggunakan nama pengguna LDAP dan kata sandi apa pun.

Mengkonfigurasi pengumpulan statistik di gudang data Teradata sumber Anda

Untuk mengonversi gudang data Teradata sumber Anda, AWS SCT gunakan statistik untuk mengoptimalkan gudang data Amazon Redshift yang telah dikonversi. Anda dapat mengumpulkan statistik AWS SCT atau mengunggah file statistik. Untuk informasi selengkapnya, lihat [Mengumpulkan atau mengunggah statistik](#).

Untuk memastikan bahwa AWS SCT dapat mengumpulkan statistik dari gudang data Anda, menyelesaikan tugas prasyarat berikut.

Untuk mengumpulkan statistik dari gudang data Teradata Anda

1. Jalankan kueri berikut untuk mengingat statistik untuk semua tabel di gudang data Anda.

```
collect summary statistics on table_name;
```

Pada contoh sebelumnya, ganti *table_name* dengan *nama* tabel sumber Anda. Ulangi kueri untuk setiap tabel yang Anda konversi.

2. Jalankan kueri berikut untuk menentukan string akun untuk pengguna, yang Anda gunakan untuk mengonversi gudang data Anda.

```
select * from dbc.accountinfo where username = 'user_name'
```

3. Aktifkan pencatatan kueri untuk pengguna tertentu menggunakan string akun dari contoh sebelumnya.

```
BEGIN QUERY LOGGING WITH OBJECTS, SQL ON ALL ACCOUNT=('M$BUSI$D$H');
```

Atau, aktifkan pencatatan kueri untuk semua pengguna database.

```
BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT SQLTEXT=0 ON ALL;
```

Setelah Anda menyelesaikan pengumpulan statistik gudang data, matikan pencatatan kueri. Untuk melakukannya, Anda dapat menggunakan contoh kode berikut.

```
end query logging with explain, objects, sql on all account=(' $M$BUSI$$D$H');
```

Mengumpulkan statistik dalam mode offline dari gudang data sumber Teradata Anda

Setelah Anda mengkonfigurasi pengumpulan statistik di gudang data Teradata Anda, Anda dapat mengumpulkan statistik dalam proyek Anda AWS SCT. Atau, Anda dapat menggunakan skrip Basic Teradata Query (BTEQ) untuk mengumpulkan statistik dalam mode offline. Kemudian, Anda dapat mengunggah file dengan statistik yang dikumpulkan ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Mengumpulkan atau mengunggah statistik](#).

Untuk mengumpulkan statistik dari gudang data Teradata Anda dalam mode offline

1. Buat `off-line_stats.bteq` skrip dengan konten berikut.

```
.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','O')
      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.dbqsqltbl 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
```

2. Buat `td_run_bteq.bat` file yang menjalankan skrip BTEQ yang Anda buat di langkah sebelumnya. Gunakan konten berikut untuk file ini.

```
@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
```

3. Buat `runme.bat` file yang menjalankan file batch yang Anda buat di langkah sebelumnya. Gunakan konten berikut untuk file ini.

```
.\td_run_bteq.bat ServerName UserName DatabaseName
```

Dalam `runme.bat` file, ganti *ServerName*, *UserName*, dan *DatabaseName* dengan nilai yang berlaku.

Kemudian, jalankan `runme.bat` file tersebut. Ulangi langkah ini untuk setiap gudang data yang Anda konversi ke Amazon Redshift.

Setelah Anda menjalankan skrip ini, Anda menerima tiga file dengan statistik untuk setiap database. Anda dapat mengunggah file-file ini ke AWS SCT proyek Anda. Untuk melakukannya, pilih gudang data Anda dari panel kiri proyek Anda, dan buka menu konteks (klik kanan). Pilih Unggah Statistik.

Pengaturan konversi Teradata ke Amazon Redshift

Untuk mengedit pengaturan konversi Teradata ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Teradata, lalu pilih Teradata — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Teradata ke Amazon Redshift.

Setelan konversi Teradata ke Amazon Redshift dalam opsi AWS SCT sertakan untuk hal berikut:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk Tambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menetapkan jumlah maksimum tabel yang AWS SCT dapat diterapkan ke klaster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke klaster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node klaster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) di Panduan Manajemen Amazon Redshift.

AWS SCT mengubah semua tabel sumber Anda, bahkan jika ini lebih dari yang dapat disimpan oleh klaster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka AWS SCT tampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke klaster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasi partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan tampilan UNION ALL dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk satu tabel sumber.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasi setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang mencakup data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke klaster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah

maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci sortir dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk kolom KEY. Anda dapat memilih opsi ini hanya jika Anda memilih opsi Gunakan pengkodean kompresi.

- Untuk menggunakan daftar eksplisit kolom dalam kode dikonversi untuk SELECT * pernyataan, pilih Gunakan deklarasi kolom eksplisit.
- Untuk meniru perilaku kunci primer dan unik di kluster Amazon Redshift Anda, pilih Emulate perilaku kunci primer dan unik.

Amazon Redshift tidak menerapkan kunci unik dan utama serta menggunakannya untuk tujuan informasi saja. Jika Anda menggunakan kendala ini dalam kode Anda, maka pastikan bahwa AWS SCT mengemulasi perilaku mereka dalam kode dikonversi.

- Untuk memastikan keunikan data dalam tabel Amazon Redshift target. Untuk melakukannya, pilih Emulate perilaku tabel SET.

Teradata membuat tabel menggunakan elemen SET sintaks sebagai pilihan default. Anda tidak dapat menambahkan duplikat baris dalam SET tabel. Jika kode sumber Anda tidak menggunakan batasan keunikan ini, maka matikan opsi ini. Dalam hal ini, kode yang dikonversi bekerja lebih cepat.

Jika kode sumber Anda menggunakan SET opsi dalam tabel sebagai batasan keunikan, aktifkan opsi ini. Dalam hal ini, AWS SCT menulis ulang INSERT . . SELECT pernyataan dalam kode dikonversi untuk meniru perilaku database sumber Anda.

Setelan pengoptimalan konversi Teradata ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Teradata ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Teradata, lalu pilih Teradata — Amazon Redshift. Di panel kiri, pilih Strategi pengoptimalan. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Teradata ke Amazon Redshift.

Setelan pengoptimalan konversi Teradata ke Amazon Redshift dalam opsi AWS SCT sertakan untuk yang berikut ini:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses penyetelan mandiri di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi pengoptimalan, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Menggunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilai (0-100). Nilai-nilai ini menentukan berat setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan memengaruhi pilihan kunci distribusi dan sortir. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Find small tables. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimalisasi konversi.

- Untuk Urutkan batas kolom kunci, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai condong lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kejadian dari nilai yang paling umum untuk jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1—100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih statistik pengguna, pilih pengguna database yang Anda ingin menganalisis statistik query.

Selain itu, pada tab Strategi pengoptimalan, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menggunakan Vertica sebagai sumber AWS SCT

Anda dapat menggunakannya AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Vertica ke Amazon Redshift.

Hak istimewa untuk Vertica sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan Vertica sebagai sumber:

- PENGGUNAAN PADA SKEMA <schema_name>
- PENGGUNAAN PADA SKEMA PUBLIK
- PILIH PADA SEMUA TABEL DALAM SKEMA <schema_name>
- PILIH PADA SEMUA URUTAN DALAM SKEMA <schema_name>
- EKSEKUSI PADA SEMUA FUNGSI DALAM SKEMA <schema_name>
- EKSEKUSI PADA PROSEDUR <schema_name.procedure_name (procedure_signature) >

Pada contoh sebelumnya, ganti placeholder sebagai berikut:

- Ganti *schema_name* dengan nama skema sumber.
- Ganti *procedure dengan nama* prosedur sumber. Ulangi hibah untuk setiap prosedur yang Anda konversi.
- Ganti *procedure signature dengan daftar dipisahkan* koma dari jenis argumen prosedur.

Menghubungkan ke Vertica sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Vertica Anda dengan file. AWS Schema Conversion Tool

Untuk menyambung ke database sumber Vertica

1. Di AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Vertica, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

3. Untuk nama Connection, masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Gunakan kredensi database dari AWS Secrets Manager atau masukkan secara manual:
 - Untuk menggunakan kredensi database dari Secrets Manager, gunakan petunjuk berikut:
 1. Untuk AWS Rahasia, pilih nama rahasianya.
 2. Pilih Mengisi untuk secara otomatis mengisi semua nilai dalam kotak dialog koneksi database dari Manajer Rahasia.

Untuk informasi tentang penggunaan kredensi database dari Secrets Manager, lihat.

[Menggunakan AWS Secrets Manager](#)

- Untuk memasukkan informasi koneksi database sumber Vertica secara manual, gunakan petunjuk berikut:

Parameter	Action
Nama server	Masukkan nama Domain Name System (DNS) atau alamat IP server database sumber Anda.

Parameter	Action
Pelabuhan server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis data	Masukkan nama database Vertica.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server database sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada tab SSL:</p> <ul style="list-style-type: none">• Verifikasi sertifikat server: Pilih opsi ini untuk memverifikasi sertifikat server menggunakan toko kepercayaan.• Toko kepercayaan: Toko kepercayaan yang Anda siapkan di pengaturan Global.• Toko kunci: Toko kunci yang Anda siapkan di pengaturan Global.
Simpan kata sandi	AWS SCT menciptakan brankas yang aman untuk menyimpan sertifikat SSL dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.

Parameter	Action
Jalur pengemudi Vertica	<p>Masukkan path ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan.</p> <p>Jika Anda menyimpan jalur driver dalam pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

- Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke database sumber Anda.
- Pilih Hubungkan untuk terhubung ke database sumber Anda.

Pengaturan konversi Vertica ke Amazon Redshift

Untuk mengedit pengaturan konversi Vertica ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Vertica, lalu pilih Vertica - Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Vertica ke Amazon Redshift.

Pengaturan konversi Vertica ke Amazon Redshift AWS SCT termasuk opsi berikut:

- Untuk membatasi jumlah komentar dengan item tindakan dalam kode yang dikonversi.

Untuk menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi, pilih tingkat keparahan item tindakan. AWS SCT menambahkan komentar dalam kode dikonversi untuk item tindakan tingkat keparahan yang dipilih dan lebih tinggi.

Misalnya, untuk meminimalkan jumlah komentar dalam kode Anda yang dikonversi, pilih Kesalahan saja. Untuk menyertakan komentar untuk semua item tindakan dalam kode yang telah dikonversi, pilih Semua pesan.

- Untuk menetapkan jumlah maksimum tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk kluster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke kluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node klaster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) di Panduan Manajemen Amazon Redshift.

AWS SCT mengubah semua tabel sumber Anda, bahkan jika ini lebih dari yang dapat disimpan oleh klaster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka AWS SCT tampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke klaster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasi partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan tampilan UNION ALL dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk satu tabel sumber.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasi setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang mencakup data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke klaster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi](#) di Panduan Pengembang Database Amazon Redshift.

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci sortir dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk kolom KEY. Anda dapat memilih opsi ini hanya jika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Vertica ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Vertica ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Vertica, lalu pilih Vertica - Amazon Redshift. Di panel kiri, pilih Strategi pengoptimalan. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Vertica ke Amazon Redshift.

Pengaturan pengoptimalan konversi Vertica ke Amazon Redshift dalam AWS SCT menyertakan opsi berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses penyetelan mandiri di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi pengoptimalan, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Menggunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilai (0-100). Nilai-nilai ini menentukan berat setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan memengaruhi pilihan kunci distribusi dan sortir. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Find small tables. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimalisasi konversi.

- Untuk Urutkan batas kolom kunci, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai condong lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kejadian dari nilai yang paling umum untuk jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1—100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih statistik pengguna, pilih pengguna database yang Anda ingin menganalisis statistik query.

Selain itu, pada tab Strategi pengoptimalan, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Max, masukkan jumlah minimum dan maksimum baris dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke meja kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Membuat aturan pemetaan di AWS SCT

Anda dapat menambahkan beberapa database sumber dan target dalam satu AWS SCT proyek. Melakukan hal ini menyederhanakan pengelolaan proyek, ketika Anda memigrasikan beberapa database ke platform target yang berbeda.

Setelah Anda membuat proyek baru dan menambahkan database sumber dan target, buat aturan pemetaan. AWS SCT memerlukan setidaknya satu aturan pemetaan untuk membuat laporan penilaian migrasi dan mengkonversi skema database.

Aturan pemetaan menjelaskan pasangan sumber-target yang mencakup skema database sumber atau database sumber dan platform database target. Anda dapat membuat beberapa aturan pemetaan dalam satu AWS SCT proyek. Gunakan aturan pemetaan untuk mengubah setiap skema database sumber ke platform database target yang tepat.

Untuk mengubah nama skema Anda dalam kode yang dikonversi, siapkan aturan migrasi. Misalnya, dengan aturan migrasi, Anda dapat mengganti nama skema Anda, menambahkan awalan ke nama objek, mengubah pemeriksaan kolom, atau mengubah tipe data. Untuk menerapkan perubahan ini ke kode yang telah dikonversi, pastikan Anda membuat aturan migrasi sebelum mengonversi skema sumber. Untuk informasi selengkapnya, lihat [Membuat aturan migrasi](#).

Anda dapat membuat aturan pemetaan hanya untuk pasangan konversi database yang didukung. Untuk daftar pasangan konversi yang didukung, lihat [Sumber untuk AWS SCT](#).

Jika Anda membuka proyek yang disimpan dalam AWS SCT versi 1.0.655 atau sebelumnya, AWS SCT secara otomatis membuat aturan pemetaan untuk semua skema database sumber ke platform database target. Untuk menambahkan platform database target lainnya, hapus aturan pemetaan yang ada, lalu buat aturan pemetaan baru.

Topik

- [Menambahkan aturan pemetaan baru](#)
- [Mengelola aturan pemetaan](#)
- [Menggunakan target virtual](#)
- [Keterbatasan untuk menggunakan beberapa server dalam satu AWS SCT proyek](#)

Menambahkan aturan pemetaan baru

Anda dapat membuat beberapa aturan pemetaan dalam satu proyek. AWS SCT menyimpan aturan pemetaan sebagai bagian dari proyek Anda. Dengan proyek Anda terbuka, gunakan prosedur berikut untuk menambahkan aturan pemetaan baru.

Untuk membuat aturan pemetaan

1. Pada menu View, pilih Tampilan pemetaan.
2. Di panel kiri, pilih skema atau database untuk ditambahkan ke aturan pemetaan.
3. Di panel kanan, pilih platform database target untuk skema sumber atau database yang dipilih.

Anda dapat memilih platform database virtual sebagai target. Untuk informasi selengkapnya, lihat [Menggunakan target virtual](#).

4. Pilih Buat pemetaan.

AWS SCT menambahkan aturan pemetaan baru ini ke daftar pemetaan Server.

Tambahkan aturan pemetaan untuk semua pasangan konversi. Untuk membuat laporan penilaian atau mengonversi skema database, pilih Tampilan utama pada menu Tampilan.

AWS SCT menyoroti dalam huruf tebal semua objek skema yang merupakan bagian dari aturan pemetaan.

Mengelola aturan pemetaan

Anda dapat memfilter atau menghapus aturan pemetaan yang ada, dan menambahkan aturan pemetaan baru di proyek AWS Schema Conversion Tool (AWS SCT) Anda.

Ketika Anda membuat aturan pemetaan untuk seluruh database sumber, AWS SCT membuat satu aturan pemetaan untuk setiap skema database sumber. Untuk proyek-proyek yang melibatkan puluhan skema atau bahkan database, mungkin sulit untuk memahami, target yang digunakan untuk skema tertentu. Untuk menemukan aturan pemetaan untuk skema Anda dengan cepat, gunakan satu atau beberapa opsi filter berikut. AWS SCT

Untuk memfilter aturan pemetaan

1. Pada menu View, pilih Tampilan pemetaan.

2. Untuk server Sumber, pilih database sumber.

Default filter adalah All, yang berarti AWS SCT menampilkan aturan pemetaan untuk semua database sumber.

3. Untuk skema Sumber, masukkan nama skema sumber. Gunakan persen (%) sebagai wildcard untuk mengganti sejumlah simbol apa pun dalam nama skema.

Default filter adalah % wildcard, yang berarti AWS SCT menampilkan aturan pemetaan untuk semua nama skema basis data sumber.

4. Untuk Memiliki aturan migrasi, pilih Ya untuk menampilkan aturan pemetaan yang membuat aturan migrasi data. Pilih Tidak untuk menampilkan aturan pemetaan yang tidak memiliki aturan migrasi data. Untuk informasi selengkapnya, lihat [Membuat aturan migrasi data di AWS SCT](#).

Default filter adalah All, yang berarti AWS SCT menampilkan semua aturan pemetaan.

5. Untuk server Target, pilih database target.

Default filter adalah All, yang berarti AWS SCT menampilkan aturan pemetaan untuk semua database target.

Dengan proyek Anda terbuka, gunakan prosedur berikut untuk menghapus aturan pemetaan.

Untuk informasi selengkapnya tentang menambahkan aturan pemetaan, lihat [Menambahkan aturan pemetaan baru](#).

Menghapus aturan pemetaan

1. Pada menu View, pilih Tampilan pemetaan.
2. Untuk pemetaan Server, pilih aturan pemetaan yang akan dihapus.
3. Pilih Hapus pemetaan yang dipilih.

AWS SCT menghapus aturan pemetaan yang dipilih.

Menggunakan target virtual

Anda dapat melihat bagaimana AWS SCT mengubah skema database sumber Anda ke platform database target yang didukung. Untuk melakukannya, Anda tidak perlu terhubung ke database target yang ada. Sebagai gantinya, Anda dapat memilih platform database target virtual di panel kanan saat Anda membuat aturan pemetaan. Untuk informasi selengkapnya, lihat [Menambahkan aturan](#)

[pemetaan baru](#). Pastikan Anda memperluas Server, kluster NoSQL, dan node ETL di panel kanan untuk melihat daftar platform database target virtual.

AWS SCT mendukung platform database target virtual berikut:

- Amazon Aurora Edisi Kompatibel MySQL
- Amazon Aurora Edisi Kompatibel PostgreSQL
- Amazon DynamoDB
- Amazon Redshift
- Amazon Redshift dan AWS Glue
- AWS Glue
- AWS Glue Studio
- Babelfish untuk Aurora PostgreSQL
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Jika Anda menggunakan Babelfish untuk Aurora PostgreSQL sebagai platform database target, Anda hanya dapat membuat laporan penilaian migrasi database. Untuk informasi selengkapnya, lihat [the section called “Laporan penilaian migrasi”](#).

Jika Anda menggunakan platform database target virtual, Anda dapat menyimpan kode yang dikonversi ke file. Untuk informasi selengkapnya, lihat [the section called “Menyimpan skema yang telah dikonversi”](#).

Keterbatasan untuk menggunakan beberapa server dalam satu AWS SCT proyek

Keterbatasan berikut berlaku ketika mengkonversi skema menggunakan beberapa server dalam satu AWS SCT proyek:

- Anda dapat menambahkan server yang sama ke proyek hanya sekali.

- Anda tidak dapat memetakan skema server ke skema target tertentu, hanya ke server target. AWS SCT menciptakan skema target selama konversi.
- Anda tidak dapat memetakan objek sumber tingkat rendah ke server target.
- Anda dapat memetakan satu skema sumber ke hanya satu server target dalam sebuah proyek.
- Pastikan untuk memetakan sumber ke server target untuk membuat laporan penilaian, mengonversi skema, atau mengekstrak data.

Membuat laporan konversi

Ketika Anda merencanakan konversi database, sangat membantu untuk membuat beberapa laporan untuk membantu Anda memahami apa yang terlibat. Anda dapat membuat laporan menggunakan AWS Schema Conversion Tool.

Anda dapat menggunakan AWS SCT untuk membuat laporan penilaian migrasi database. Dengan laporan ini, Anda mendapatkan ringkasan tugas konversi skema Anda dan rincian untuk item yang tidak dapat secara otomatis dikonversi ke database target Anda. Anda dapat menggunakan laporan ini untuk mengevaluasi berapa banyak proyek dapat diselesaikan dengan menggunakan AWS SCT, dan apa lagi yang Anda butuhkan untuk menyelesaikan konversi. Untuk membuat laporan penilaian, gunakan **Buat Laporan** dari menu konteks (klik kanan) dari database di AWS SCT.

Topik

- [Membuat laporan penilaian migrasi dengan AWS SCT](#)

Membuat laporan penilaian migrasi dengan AWS SCT

Bagian penting dari AWS Schema Conversion Tool adalah laporan penilaian yang dihasilkannya untuk memperkirakan kompleksitas konversi skema Anda. Laporan penilaian migrasi database ini merangkum semua tugas konversi skema dan merinci item tindakan untuk skema yang tidak dapat dikonversi ke mesin DB instans DB target Anda. Anda dapat melihat laporan dalam aplikasi atau mengekspornya sebagai file PDF (comma-separated value (CSV) atau PDF).

Jika Anda menambahkan beberapa database sumber dan target dalam satu proyek, AWS SCT agregat laporan untuk semua pasangan konversi menjadi satu laporan penilaian migrasi database.

Anda dapat menggunakan platform database target virtual untuk membuat laporan penilaian dan memahami kompleksitas migrasi ke platform database yang dipilih. Dalam hal ini, Anda tidak perlu terhubung ke platform basis data target basis data Anda. Misalnya, Anda dapat menggunakan Babelfish untuk Aurora PostgreSQL sebagai platform database target virtual untuk membuat laporan penilaian migrasi database. Untuk informasi lebih lanjut tentang platform database target virtual, lihat [the section called "Target virtual"](#).

Laporan penilaian migrasi mencakup hal-hal berikut:

- Ringkasan eksekutif
- evaluasi lisensi

- Dukungan cloud, menunjukkan fitur apa pun dalam database sumber tidak tersedia pada target.
- Rekomendasi, termasuk konversi objek server, saran cadangan, dan perubahan server terkait

Laporan ini juga mencakup perkiraan jumlah upaya yang diperlukan untuk menulis kode setara untuk instans DB target Anda yang tidak dapat dikonversi secara otomatis.

Jika Anda menggunakan AWS SCT untuk memigrasi skema yang ada ke instans DB Amazon RDS, maka Anda dapat menggunakan laporan tersebut untuk membantu menganalisis persyaratan untuk pindah ke AWS Cloud dan mengubah jenis lisensi Anda.

Topik

- [Membuat laporan penilaian penilaian penilaian penilaian migrasi basis data penilaian migrasi basis data](#)
- [Melihat laporan penilaian](#)
- [Menyimpan laporan penilaian penilaian penilaian penilaian penilaian penilaian penilaian penilaian](#)
- [Mengonfigurasi laporan penilaian penilaian penilaian penilaian penilaian penilaian penilaian penilaian penilaian penilaian](#)
- [Membuat laporan penilaian multiserver untuk migrasi database](#)

Membuat laporan penilaian penilaian penilaian penilaian migrasi basis data penilaian migrasi basis data

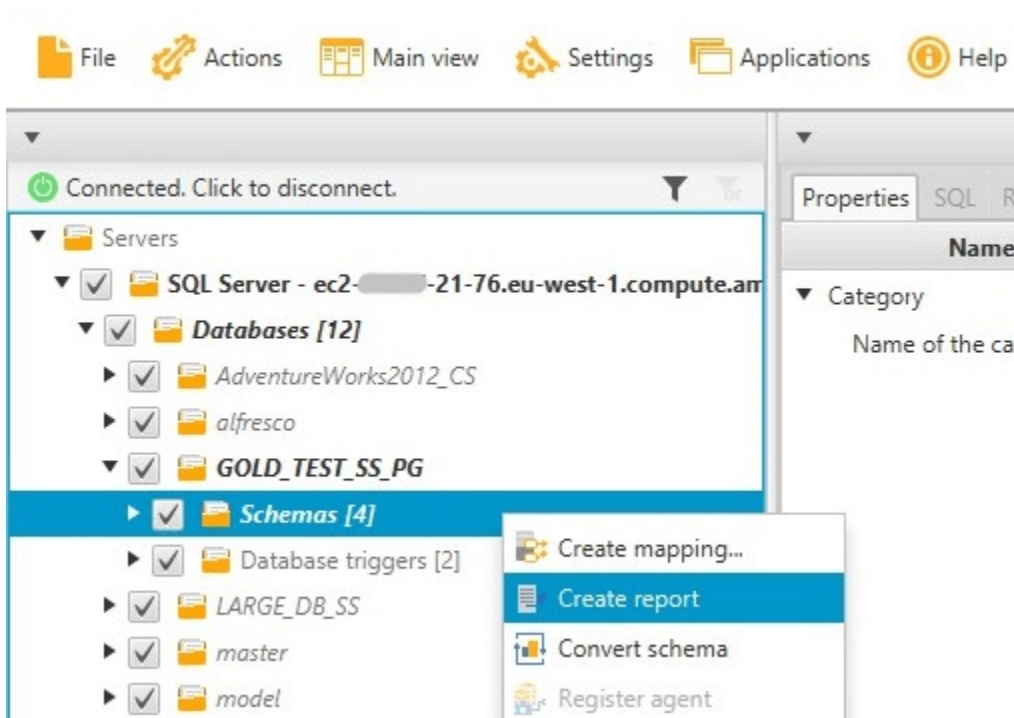
Gunakan prosedur berikut untuk membuat laporan penilaian penilaian penilaian penilaian migrasi basis data.

Membuat laporan penilaian migrasi database

1. Pastikan Anda membuat aturan pemetaan untuk skema database sumber untuk membuat laporan penilaian. Untuk informasi selengkapnya, lihat [Menambahkan aturan pemetaan baru](#).
2. Pada menu Lihat, pilih Tampilan utama.
3. Di panel kiri yang menampilkan skema basis data sumber data sumber Anda, memilih objek skema untuk membuat laporan penilaian penilaian penilaian penilaian untuk. Untuk menyertakan beberapa skema database ke dalam laporan, pilih node induk, misalnya Skema.

Pastikan Anda memilih kotak centang untuk semua objek skema yang akan dibuat laporan penilaian.

4. Buka menu konteks (klik kanan) untuk objek, lalu memilih Buat laporan konteks (klik kanan).



Melihat laporan penilaian

Setelah Anda membuat laporan penilaian, tampilan laporan penilaian akan terbuka, menampilkan tab berikut:

- Ringkasan
- Item Tindakan

Itu Ringkasan tab menunjukkan item yang secara otomatis dikonversi atau tidak dikonversi.

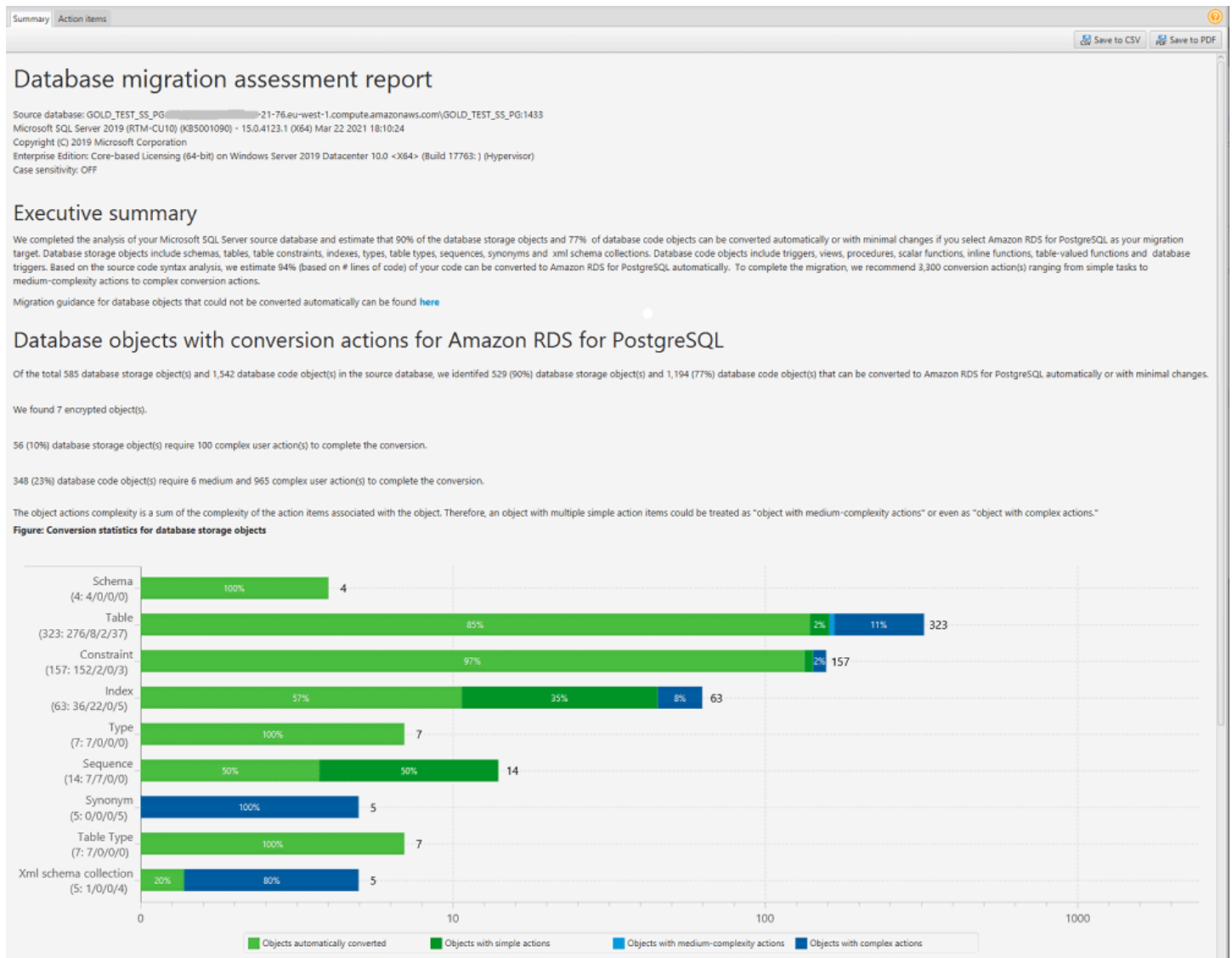
Tab Item Tindakan menampilkan item yang tidak dapat dikonversi secara otomatis, dan rekomendasi tentang apa yang harus dilakukan terhadapnya.

Topik

- [Ringkasan laporan penilaian penilaian penilaian penilaian penilaian](#)
- [Item tindakan laporan penilaian](#)
- [Pesan peringatan laporan penilaian](#)

Ringkasan laporan penilaian penilaian penilaian penilaian penilaian

Tab Ringkasan menampilkan informasi ringkasan dari laporan penilaian migrasi database. Ini menunjukkan item yang dikonversi secara otomatis, dan item yang tidak dikonversi secara otomatis.



Untuk item skema yang tidak dapat dikonversi secara otomatis ke mesin database target, ringkasan mencakup perkiraan upaya yang diperlukan untuk membuat item skema dalam instans DB target Anda yang setara dengan yang ada di sumber Anda.

Laporan ini mengkategorikan perkiraan waktu untuk mengonversi item skema ini sebagai berikut:

- Sederhana - Tindakan yang dapat diselesaikan dalam waktu kurang dari dua jam.
- Medium - Tindakan yang lebih kompleks dan dapat diselesaikan dalam dua hingga enam jam.

- Signifikan - Tindakan yang sangat kompleks dan membutuhkan waktu lebih dari enam jam untuk menyelesaikannya.

Bagian Evaluasi Lisensi dan Support Cloud berisi informasi tentang memindahkan skema database lokal yang ada ke instans DB Amazon RDS yang menjalankan mesin yang sama. Misalnya, jika Anda ingin mengubah jenis lisensi, bagian laporan ini memberi tahu Anda fitur mana dari database Anda saat ini yang harus dihapus.

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.

Item tindakan laporan penilaian

Tampilan laporan penilaian juga mencakup tab Item Tindakan. Tab ini berisi daftar item yang tidak dapat dikonversi secara otomatis ke mesin database instans DB Amazon RDS target Anda. Jika Anda memilih item tindakan dari daftar, AWS SCT sorot item dari skema Anda yang berlaku untuk item tindakan.

Laporan ini juga berisi rekomendasi tentang cara mengonversi item skema secara manual. Misalnya, setelah penilaian berjalan, laporan terperinci untuk database/skema menunjukkan kepada Anda upaya yang diperlukan untuk merancang dan mengimplementasikan rekomendasi untuk mengonversi item Tindakan. Untuk informasi selengkapnya tentang memutuskan cara menangani konversi manual, lihat [Menangani konversi manual AWS SCT](#).

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the source database structure, including servers, databases, tables, and schemas. The main pane shows a list of action items (issues) with their recommended actions and occurrence counts. Below this, a SQL procedure definition is shown for the target Amazon RDS MySQL instance.

Group by: Issues | **Complexity:** All | **Object type:** All

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 | Documentation reference(s): <http://dev.mysql.com/doc/refman/8.0/en/insert.html>
- 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
Parameter: @InputPosNo (Number of occurrences: 1)
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**
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 | Documentation reference(s): <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
- 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

Source: Microsoft SQL Server procedure: POSITION_UPDATE_CASH_CGT_BULK

```

1 create procedure POSITION_UPDATE_CASH_CGT_BULK
2   @InputPosNo tinyint(4) 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

Name	Value
Category	Schemas
Name of the category	Schemas

Pesan peringatan laporan penilaian

Untuk menilai kompleksitas mengkonversi ke mesin database lain, AWS SCT memerlukan akses ke objek dalam database sumber Anda. Ketika SCT tidak dapat melakukan penilaian karena masalah ditemui selama pemindaian, pesan peringatan dikeluarkan yang menunjukkan persentase konversi keseluruhan berkurang.

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:

- 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.

Berikut ini adalah alasan mengapa AWS SCT mungkin mengalami masalah selama pemindaian:

- Akun pengguna yang terhubung ke database tidak memiliki akses ke semua objek yang dibutuhkan.
- Objek yang dikutip dalam skema tidak ada lagi dalam database.

- SCT sedang mencoba menilai objek yang dienkripsi.

Untuk informasi selengkapnya tentang izin keamanan dan hak istimewa SCT yang diperlukan untuk database Anda, lihat [Sumber untuk AWS SCT](#) bagian database sumber yang sesuai dalam panduan ini.

Menyimpan laporan penilaian penilaian penilaian penilaian penilaian penilaian penilaian penilaian

Setelah [membuat laporan penilaian migrasi database](#), Anda dapat menyimpan salinan lokal laporan penilaian migrasi database sebagai file PDF atau file nilai dipisahkan koma (CSV).

Untuk menyimpan laporan penilaian migrasi database sebagai file PDF

1. Di menu atas, pilih Lihat, lalu pilih Tampilan laporan penilaian.
2. Pilih tab Ringkasan.
3. Pilih Simpan ke PDF di kanan atas.

Untuk menyimpan laporan penilaian migrasi database sebagai file CSV

1. Di menu atas, pilih Lihat, lalu pilih Tampilan laporan penilaian.
2. Pilih tab Ringkasan.
3. Pilih Simpan ke CSV di kanan atas.

File PDF berisi informasi ringkasan dan item tindakan item ringkasan dan tindakan, seperti yang ditunjukkan pada contoh berikut.

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.

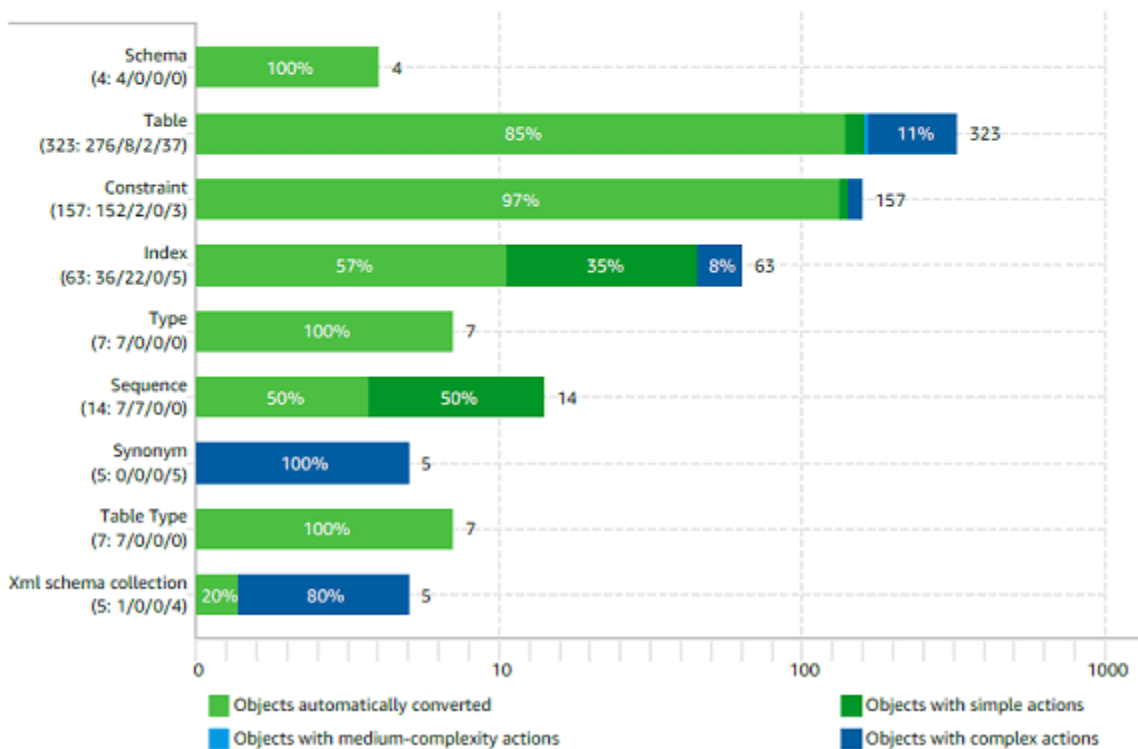
We found 7 encrypted object(s).

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



Saat Anda memilih opsi Simpan ke CSV, AWS SCT buat tiga file CSV.

File CSV pertama berisi informasi berikut tentang item tindakan berikut:

- Kategori
- Kejadian - Nama file, nomor baris, dan posisi untuk item
- Nomor item tindakan
- Subjek
- Grup

- Deskripsi
- Referensi dokumentasi
- Tindakan yang direkomendasikan
- Kompleksitas yang diperkirakan

File CSV kedua menyertakan `Action_Items_Summary` akhiran dalam namanya dan berisi informasi tentang jumlah kejadian semua item tindakan.

Pada contoh berikut, nilai dalam kolom Upaya kurva pembelajaran menunjukkan jumlah upaya yang diperlukan untuk merancang pendekatan untuk mengonversi setiap item tindakan. Nilai dalam Upaya mengonversi kejadian kolom item tindakan menunjukkan upaya yang diperlukan untuk mengonversi setiap item tindakan, mengikuti pendekatan yang dirancang. Nilai yang digunakan untuk menunjukkan tingkat usaha yang dibutuhkan didasarkan pada skala tertimbang, mulai dari rendah (paling sedikit) hingga tinggi (paling).

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

File CSV ketiga termasuk `Summary` dalam namanya dan berisi ringkasan berikut:

- Kategori
- Jumlah objek
- Objek otomatis dikonversi
- Objek dengan tindakan sederhana
- Objek dengan tindakan kompleksitas sedang
- Objek dengan tindakan kompleks
- Total baris kode

Database objects with conversion actions for Amazon RDS for PostgreSQL

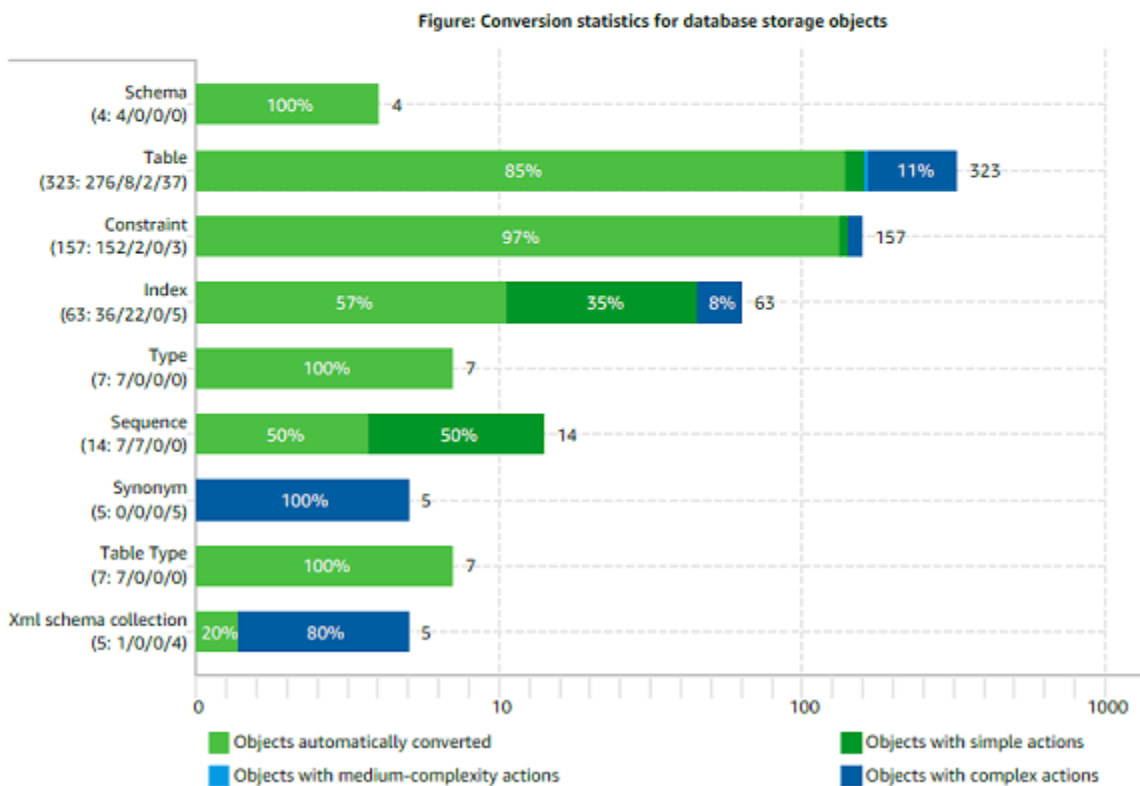
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

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."



Saat Anda memilih opsi Simpan ke CSV, AWS SCT buat tiga file CSV.

File CSV pertama berisi informasi berikut tentang item tindakan berikut:

- Kategori
- Kejadian - Nama file, nomor baris, dan posisi untuk item
- Nomor item tindakan
- Subjek
- Grup

- Deskripsi

- Referensi dokumentasi

- Tindakan yang direkomendasikan

- Kompleksitas yang diperkirakan

File CSV kedua menyertakan `Action_Items_Summary` akhiran dalam namanya dan berisi informasi tentang jumlah kejadian semua item tindakan.

Pada contoh berikut, nilai dalam kolom Upaya kurva pembelajaran menunjukkan jumlah upaya yang diperlukan untuk merancang pendekatan untuk mengonversi setiap item tindakan. Nilai dalam Upaya mengonversi kejadian kolom item tindakan menunjukkan upaya yang diperlukan untuk mengonversi setiap item tindakan, mengikuti pendekatan yang dirancang. Nilai yang digunakan untuk menunjukkan tingkat usaha yang dibutuhkan didasarkan pada skala tertimbang, mulai dari rendah (paling sedikit) hingga tinggi (paling).

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

File CSV ketiga termasuk `Summary` dalam namanya dan berisi ringkasan berikut:

- Kategori

- Jumlah objek

- Objek otomatis dikonversi

- Objek dengan tindakan sederhana

- Objek dengan tindakan kompleksitas sedang

- Objek dengan tindakan kompleks

- Total baris kode

Membuat laporan penilaian multiserver untuk migrasi database

Untuk menentukan arah target terbaik untuk lingkungan Anda secara keseluruhan, buat laporan penilaian multiserver.

Laporan penilaian multiserver mengevaluasi beberapa server berdasarkan masukan yang Anda berikan untuk setiap definisi skema yang ingin Anda nilai. Definisi skema Anda berisi parameter koneksi server database dan nama lengkap dari setiap skema. Setelah menilai setiap skema, buatlah AWS SCT ringkasan, laporan penilaian agregat untuk migrasi database di beberapa server Anda. Laporan ini menunjukkan perkiraan kompleksitas untuk setiap target migrasi yang mungkin.

Anda dapat menggunakan AWS SCT untuk membuat laporan penilaian penilaian penilaian multiserver untuk sumber dan target basis data penilaian multiserver berikut.

Basis data sumber data sumber daya	Target database
Amazon Redshift	Amazon Redshift
Basis data Azure SQL	Aurora MySQL, PostgreSQL, PostgreSQL
Analytics Azure Synapse	Amazon Redshift
BigQuery	Amazon Redshift
Greenplum	Amazon Redshift
IBM Db2 untuk z/OS	Amazon Aurora Amazon Aurora Amazon Aurora Kompatibel PostgreSQL (Aurora PostgreSQL)
IBM Db2 LUW	Aurora MySQL, PostgreSQL MariaDB PostgreSQL, PostgreSQL
Microsoft SQL Server	Aurora MySQL, PostgreSQL, PostgreSQL, Babelfish untuk Aurora PostgreSQL, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL	Aurora PostgreSQL, MySQL, PostgreSQL
Netezza	Amazon Redshift

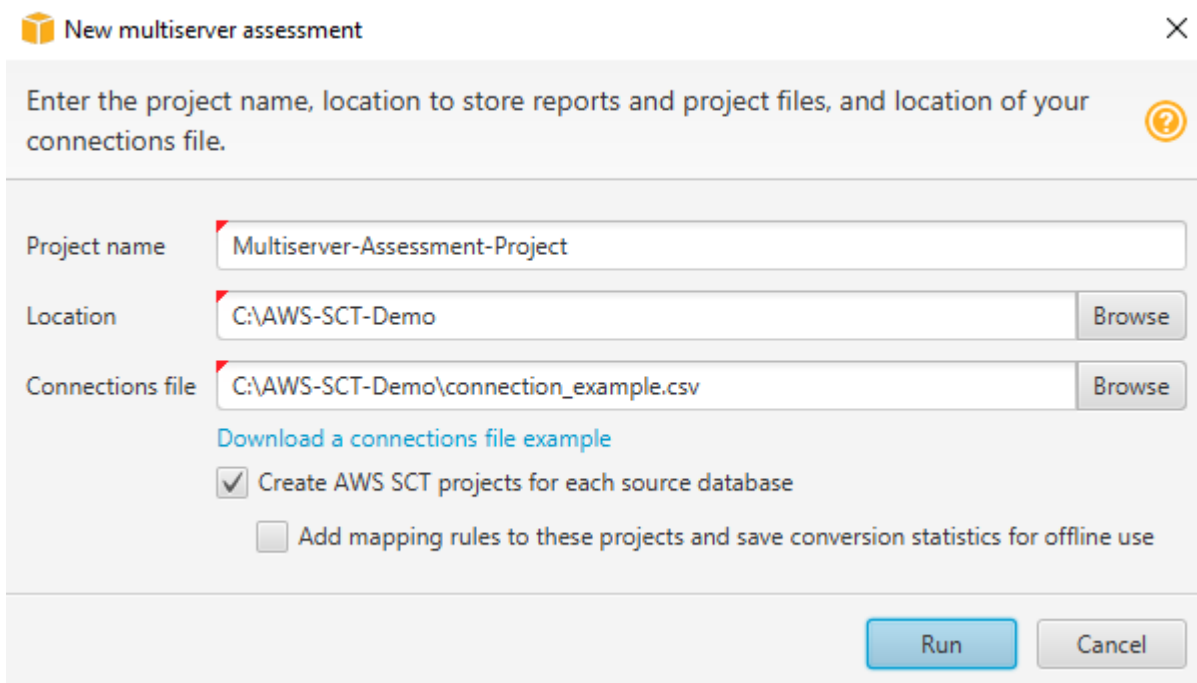
Basis data sumber data sumber daya	Target database
Oracle	Aurora MySQL, PostgreSQL, PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL	Aurora MySQL, PostgreSQL, PostgreSQL
SAP ASE	Aurora MySQL, PostgreSQL MariaDB PostgreSQL, PostgreSQL
Kepingan salju	Amazon Redshift
Teradata	Amazon Redshift
Vertica	Amazon Redshift

Melakukan penilaian multiserver

Gunakan prosedur berikut untuk melakukan penilaian multiserver dengan AWS SCT. Anda tidak perlu membuat proyek baru AWS SCT untuk melakukan penilaian multiserver. Sebelum memulai, pastikan Anda telah menyiapkan file comma-separated value (CSV) dengan parameter koneksi basis data. Juga, pastikan Anda telah menginstal semua driver database yang diperlukan dan mengatur lokasi driver dalam AWS SCT pengaturan. Untuk informasi selengkapnya, lihat [Mengunduh driver database yang diperlukan](#).

Untuk melakukan penilaian multiserver dan membuat laporan ringkasan gabungan

1. Di AWS SCT, pilih File, Penilaian multiserver baru. Kotak dialog penilaian multiserver baru terbuka.



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. Pilih Unduh contoh file koneksi untuk mengunduh template kosong file CSV dengan parameter koneksi database.
3. Masukkan nilai untuk nama Proyek, Lokasi (untuk menyimpan laporan), dan file Koneksi (file CSV).
4. Pilih Buat AWS SCT proyek untuk setiap database sumber untuk membuat proyek migrasi secara otomatis setelah membuat laporan penilaian.
5. Dengan Membuat AWS SCT proyek untuk setiap database sumber diaktifkan, Anda dapat memilih Tambahkan aturan pemetaan ke proyek ini dan menyimpan statistik konversi untuk penggunaan offline. Dalam hal ini, AWS SCT akan menambahkan aturan pemetaan untuk setiap proyek dan menyimpan metadata database sumber dalam proyek. Untuk informasi selengkapnya, lihat [Berjalan AWS SCT dalam mode offline](#).

6. Memilih Jalankan.

Sebuah progress bar muncul menunjukkan kecepatan penilaian database. Jumlah mesin target dapat mempengaruhi runtime penilaian.

7. Pilih Ya jika pesan berikut ditampilkan: Analisis penuh dari semua server Database mungkin memakan waktu lama. Apakah Anda ingin melanjutkan?

Ketika laporan penilaian multiserver selesai, layar muncul menunjukkan demikian.

8. Pilih Buka Laporan untuk melihat laporan penilaian ringkasan agregat.

Secara default,AWS SCT menghasilkan laporan gabungan untuk semua database sumber dan laporan penilaian terperinci untuk setiap nama skema dalam database sumber. Untuk informasi selengkapnya, lihat [Menemukan dan melihat laporan](#).

Dengan opsi BuatAWS SCT proyek untuk setiap basis data sumber diaktifkan,AWS SCT buat proyek kosong untuk setiap database sumber. AWS SCTjuga membuat laporan penilaian penilaian penilaian penilaian seperti yang dijelaskan sebelumnya. Setelah Anda menganalisis laporan penilaian ini dan memilih tujuan migrasi untuk setiap database sumber, tambahkan database target ke proyek kosong ini.

Dengan opsi Tambahkan aturan pemetaan ke proyek ini dan simpan statistik konversi untuk penggunaan offline diaktifkan,AWS SCT buat proyek untuk setiap database sumber. Proyek-proyek ini mencakup informasi berikut:

- Database sumber Anda dan platform database target virtual. Untuk informasi selengkapnya, lihat [Menggunakan target virtual](#).
- Aturan pemetaan untuk pasangan target sumber ini. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan](#).
- Laporan penilaian migrasi database untuk pasangan target sumber ini.
- Sumber skema metadata, yang memungkinkan Anda untuk menggunakanAWS SCT proyek ini dalam mode offline. Untuk informasi selengkapnya, lihat [BerjalanAWS SCT dalam mode offline](#).

Mempersiapkan file CSV masukan

Untuk memberikan parameter koneksi sebagai masukan untuk laporan penilaian penilaian penilaian multiserver, gunakan file CSV seperti yang ditunjukkan dalam contoh berikut.

```
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,,,,,AURORA
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
```

Contoh sebelumnya menggunakan titik koma untuk memisahkan dua nama skema untukSales database. Hal ini juga menggunakan titik koma untuk memisahkan dua target platform migrasi database untukSales database.

Juga, contoh sebelumnya menggunakanAWS Secrets Manager untuk terhubung keCustomers database dan Windows Authentication untuk terhubung keHR database.


Anda dapat membuat file CSV baru atau mengunduh template untuk file CSV dariAWS SCT dan mengisi informasi yang diperlukan. Pastikan baris pertama file CSV Anda menyertakan nama kolom yang sama seperti yang ditunjukkan pada contoh sebelumnya.

Untuk mengunduh template file CSV masukan

1. Mulai AWS SCT.
2. Pilih File, lalu pilih Penilaian multiserver baru.
3. Pilih Unduh contoh file koneksi.

Pastikan file CSV Anda menyertakan nilai berikut, yang disediakan oleh template:

- Nama - Label teks yang membantu mengidentifikasi database Anda. AWS SCTmenampilkan label teks ini dalam laporan penilaian.
- Deskripsi - Nilai opsional, di mana Anda dapat memberikan informasi tambahan tentang database.
- Secret Manager Key - Nama rahasia yang menyimpan kredensi database Anda diAWS Secrets Manager. Untuk menggunakan Secrets Manager, pastikan Anda menyimpanAWS profilAWS SCT. Untuk informasi selengkapnya, lihat [Menggunakan AWS Secrets Manager](#).

 Important

AWS SCTmengabaikan parameter Kunci Manajer Rahasia jika Anda menyertakan parameter IP Server, Port, Login, dan Kata Sandi dalam file input.

- IP server — Nama Domain Name Service (DNS) atau alamat IP server basis data sumber Anda.
- Port - Port yang digunakan untuk terhubung ke server database sumber Anda.
- Nama Layanan - Jika Anda menggunakan nama layanan untuk terhubung ke database Oracle Anda, nama layanan Oracle untuk terhubung ke.
- Nama database - Nama database. Untuk basis data Oracle, gunakan ID Oracle System (SID).

- BigQuery path — path ke file kunci akun layanan untuk BigQuery database sumber Anda. Untuk informasi selengkapnya tentang membuat file ini, lihat [Hak istimewa untuk BigQuery sebagai sumber](#).
- Mesin Sumber - Jenis database sumber Anda. Gunakan salah satu nilai berikut:
 - AZURE_MSSQL untuk Database Azure SQL.
 - AZURE_SYNAPSE untuk database Azure Synapse Analytics.
 - GOOGLE_BIGQUERY untuk BigQuery database.
 - DB2ZOS untuk IBM Db2 untuk database z/OS.
 - DB2LUW untuk basis data IBM Db2 LUW.
 - GREENPLUM untuk database Greenplum.
 - MSSQL untuk basis data Microsoft SQL Server.
 - MYSQL untuk database MySQL.
 - NETEZZA untuk database Netezza.
 - ORACLE untuk database Oracle.
 - POSTGRESQL untuk database PostgreSQL.
 - REDSHIFT untuk basis data Amazon Redshift.
 - SNOWFLAKE untuk database Snowflake.
 - SYBASE_ASE untuk database SAP ASE.
 - TERADATA untuk database Teradata.
 - VERTICA untuk database Vertica.
- Nama skema - Nama-nama skema database untuk dimasukkan dalam laporan penilaian.

Untuk Azure SQL Database, Azure Synapse Analytics BigQuery,, Netezza, SAP ASE, Snowflake, dan SQL Server, gunakan format berikut nama skema:

db_name.schema_name

Pasang *db_name* kembali dengan nama basis data sumber data sumber sumber.

Pasang *schema_name* kembali dengan nama skema sumber sumber.

Lampirkan database atau skema nama yang menyertakan titik dalam tanda kutip ganda seperti yang ditunjukkan berikut: "database.name"."schema.name".

Pisahkan beberapa nama skema dengan menggunakan titik koma seperti yang ditunjukkan berikut:Schema1;Schema2.

Nama basis data dan skema peka terhadap huruf besar/kecil.

Gunakan persen (%) sebagai wildcard untuk mengganti sejumlah simbol apa pun dalam database atau nama skema. Contoh sebelumnya menggunakan persen (%) sebagai wildcard untuk menyertakan semua skema dariemployees database dalam laporan penilaian.

- Gunakan otentikasi Windows - Jika Anda menggunakan otentikasi Windows untuk menyambung ke database Microsoft SQL Server Anda, masukkan benar. Untuk informasi selengkapnya, lihat [Menggunakan Windows Authentication saat menggunakan Microsoft SQL Server sebagai sumber](#).
- Login - Nama pengguna untuk terhubung ke server basis data sumber data sumber Anda.
- Kata Sandi - Kata sandi untuk terhubung ke server database sumber Anda.
- Gunakan SSL - Jika Anda menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database sumber Anda, masukkan true.
- Toko kepercayaan - Toko kepercayaan yang akan digunakan untuk koneksi SSL Anda.
- Toko kunci - Toko kunci yang digunakan untuk koneksi SSL Anda.
- Otentikasi SSL - Jika Anda menggunakan otentikasi SSL berdasarkan sertifikat, masukkan true.
- Mesin Target - Platform basis data target. Gunakan nilai berikut untuk menentukan satu atau beberapa target dalam laporan penilaian:
 - AURORA_MYSQL untuk database yang kompatibel dengan Aurora MySQL.
 - AURORA_POSTGRES SQL untuk database yang kompatibel dengan Aurora PostgreSQL.
 - BABELFISH untuk Babelfish untuk database Aurora PostgreSQL.
 - MARIA_DB untuk database MariaDB.
 - MSSQL untuk basis data Microsoft SQL Server.
 - MYSQL untuk database MySQL.
 - ORACLE untuk database Oracle.
 - POSTGRES SQL untuk database PostgreSQL.
 - REDSHIFT untuk basis data Amazon Redshift.

Pisahkan beberapa target dengan menggunakan titik koma seperti ini:MYSQL;MARIA_DB. Jumlah target mempengaruhi waktu yang dibutuhkan untuk menjalankan penilaian.

Menemukan dan melihat laporan

penilaian multiserver menghasilkan dua jenis laporan:

- Laporan agregat dari semua database sumber.
- Sebuah laporan penilaian rinci database target untuk setiap nama skema dalam database sumber.

Laporan disimpan di direktori yang Anda pilih untuk Lokasi di kotak dialog Penilaian multiserver baru.

Untuk mengakses laporan rinci, Anda dapat menavigasi subdirektori, yang diatur oleh database sumber, nama skema, dan mesin database target.

Laporan agregat menunjukkan informasi dalam empat kolom tentang kompleksitas konversi database target. Kolom mencakup informasi tentang konversi objek kode, objek penyimpanan, elemen sintaks, dan kompleksitas konversi.

Contoh berikut menunjukkan informasi untuk konversi dua skema database Oracle ke 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

Empat kolom yang sama ditambahkan ke laporan untuk setiap mesin database target tambahan yang ditentukan.

Untuk detail tentang cara membaca informasi ini, lihat berikut.

Output untuk laporan penilaian agregat

Laporan penilaian migrasi database multiserver agregat AWS Schema Conversion Tool adalah file CSV dengan kolom berikut:

- 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*

Untuk mengumpulkan informasi, AWS SCT jalankan laporan penilaian lengkap dan kemudian agregat laporan berdasarkan skema.

Dalam laporan tersebut, tiga bidang berikut menunjukkan persentase kemungkinan konversi otomatis berdasarkan penilaian:

Konversi objek kode%

Persentase objek kode dalam skema yang AWS SCT dapat mengkonversi secara otomatis atau dengan perubahan minimal. Objek kode meliputi prosedur, fungsi, tampilan, dan sejenisnya.

Konversi objek penyimpanan%

Persentase objek penyimpanan yang SCT dapat mengkonversi secara otomatis atau dengan perubahan minimal. Objek penyimpanan termasuk tabel, indeks, kendala, dan sejenisnya.

Konversi elemen sintaks%

Persentase elemen sintaks yang SCT dapat mengkonversi secara otomatis. Elemen sintaks meliputi `SELECTFROM`, `DELETE`, dan `JOIN` klausa, dan serupa.

Perhitungan kompleksitas konversi didasarkan pada gagasan item tindakan. Item tindakan mencerminkan jenis masalah yang ditemukan dalam kode sumber yang perlu Anda perbaiki secara manual selama migrasi ke target tertentu. Item tindakan dapat memiliki beberapa kejadian.

Skala tertimbang mengidentifikasi tingkat kompleksitas untuk melakukan migrasi. Angka 1 mewakili tingkat kompleksitas terendah, dan angka 10 mewakili tingkat kompleksitas tertinggi.

Mengonversi skema database menggunakan AWS SCT

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengkonversi skema database yang ada dari satu mesin database yang lain. Mengkonversi database menggunakan antarmuka AWS SCT pengguna dapat cukup sederhana, tetapi ada beberapa hal yang perlu dipertimbangkan sebelum Anda melakukan konversi.

Misalnya, Anda dapat menggunakan AWS SCT untuk melakukan hal berikut:

- Anda dapat menggunakan AWS SCT untuk menyalin skema database lokal yang ada ke instans DB Amazon RDS yang menjalankan mesin yang sama. Anda dapat menggunakan fitur ini untuk menganalisis potensi penghematan biaya pindah ke cloud dan mengubah jenis lisensi Anda.
- Dalam beberapa kasus, fitur database tidak dapat dikonversi ke fitur Amazon RDS yang setara. Jika Anda meng-host dan mengelola sendiri database di platform Amazon Elastic Compute Cloud (Amazon EC2), Anda dapat meniru fitur-fitur ini dengan mengganti AWS layanan untuk mereka.
- AWS SCT mengotomatiskan sebagian besar proses konversi skema database pemrosesan transaksi online (OLTP) Anda ke instans MySQL DB Amazon Relational Database Service (Amazon RDS), klaster DB Amazon Aurora, atau instans DB PostgreSQL. Mesin database sumber dan target berisi banyak fitur dan kemampuan yang berbeda, dan AWS SCT upaya untuk membuat skema yang setara dalam instans DB Amazon RDS Anda sedapat mungkin. Jika tidak ada konversi langsung yang mungkin, AWS SCT berikan daftar tindakan yang mungkin untuk Anda ambil.

Topik

- [Membuat aturan migrasi di AWS SCT](#)
- [Mengonversi skema Anda dengan menggunakan AWS SCT](#)
- [Menangani konversi manual AWS SCT](#)
- [Memperbarui dan menyegarkan skema yang telah dikonversi AWS SCT](#)
- [Menyimpan dan menerapkan skema yang telah dikonversi AWS SCT](#)
- [Membandingkan skema database](#)
- [Menemukan objek yang ditransformasikan terkait](#)

AWS SCT mendukung konversi pemrosesan transaksi online (OLTP) berikut.

Basis data sumber	Target database
IBM Db2 untuk z/OS (versi 12)	Edisi Kompatibel dengan Amazon Aurora MySQL, Edisi Kompatibel dengan Amazon Aurora PostgreSQL, MySQL, PostgreSQL
IBM Db2 LUW (versi 9.1, 9.5, 9.7, 10.5, 11.1, dan 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Database Microsoft Azure SQL	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
Microsoft SQL Server (versi 2008 R2 dan yang lebih tinggi)	Aurora MySQL, Aurora PostgreSQL, Babelfish untuk Aurora PostgreSQL, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL (versi 5.5 dan lebih tinggi)	Aurora PostgreSQL, MySQL, PostgreSQL Anda dapat memigrasikan skema dan data dari MySQL ke cluster Aurora MySQL DB tanpa menggunakan AWS SCT Untuk informasi selengkapnya, lihat Memigrasi data ke kluster DB Amazon Aurora .
Oracle (versi 10.2 dan lebih tinggi)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL (versi 9.1 dan lebih tinggi)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
SAP ASE (12.5, 15.0, 15.5, 15.7, dan 16.0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL

Untuk informasi tentang mengonversi skema gudang data, lihat. [Mengonversi skema gudang data ke Amazon Redshift menggunakan AWS SCT](#)

Untuk mengonversi skema database Anda ke Amazon RDS, Anda mengambil langkah-langkah tingkat tinggi berikut:

- [Membuat aturan migrasi di AWS SCT](#)- Sebelum Anda mengonversi skema Anda AWS SCT, Anda dapat mengatur aturan yang mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek.
- [Mengonversi skema Anda dengan menggunakan AWS SCT](#)- AWS SCT membuat versi lokal dari skema yang dikonversi untuk Anda tinjau, tetapi itu tidak menerapkannya ke instans DB target Anda sampai Anda siap.
- [Membuat laporan penilaian migrasi dengan AWS SCT](#)- AWS SCT membuat laporan penilaian migrasi database yang merinci elemen skema yang tidak dapat dikonversi secara otomatis. Anda dapat menggunakan laporan ini untuk mengidentifikasi di mana Anda perlu membuat skema dalam instans DB Amazon RDS yang kompatibel dengan database sumber Anda.
- [Menangani konversi manual AWS SCT](#)- Jika Anda memiliki elemen skema yang tidak dapat dikonversi secara otomatis, Anda memiliki dua pilihan: memperbarui skema sumber dan kemudian mengonversi lagi, atau membuat elemen skema yang setara dalam instans DB Amazon RDS target Anda.
- [Memperbarui dan menyegarkan skema yang telah dikonversi AWS SCT](#)- Anda dapat memperbarui AWS SCT proyek Anda dengan skema terbaru dari database sumber Anda.
- [Menyimpan dan menerapkan skema yang telah dikonversi AWS SCT](#)- Saat Anda siap, AWS SCT terapkan skema yang dikonversi dalam proyek lokal Anda ke instans DB Amazon RDS target Anda.

Membuat aturan migrasi di AWS SCT

Sebelum mengonversi skema AWS SCT, Anda dapat mengatur aturan migrasi. Aturan migrasi di AWS SCT dapat melakukan transformasi seperti mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek. Misalnya, Anda memiliki satu set tabel dalam skema sumber Anda bernama `test_TABLE_NAME`. Anda dapat mengatur aturan yang mengubah awalan `test_` ke awalan `demo_` dalam skema target.

Note

Anda hanya dapat membuat aturan migrasi untuk mesin database sumber dan target yang berbeda.

Anda dapat membuat aturan migrasi yang melakukan tugas-tugas berikut:

- Menambahkan, menghapus, atau mengganti awalan
- Menambahkan, menghapus, atau mengganti akhiran
- Ubah kolasi kolom
- Ubah tipe data
- Mengubah panjang `char`, `varchar`, `nvarchar`, dan tipe `string` data
- Pindahkan objek
- Ubah nama objek

Anda dapat membuat aturan migrasi untuk objek berikut:

- Basis Data
- Skema
- Tabel
- Kolom

Membuat aturan migrasi

Anda dapat membuat aturan migrasi dan menyimpan aturan sebagai bagian dari proyek Anda. Dengan proyek Anda terbuka, gunakan prosedur berikut untuk membuat aturan migrasi.

Membuat aturan migrasi

1. Pada menu View, pilih Tampilan pemetaan.
2. Di Pemetaan server, pilih sepasang server sumber dan target.
3. Pilih Aturan migrasi baru. Kotak dialog Transformation rules muncul.
4. Pilih Tambahkan aturan baru. Baris baru ditambahkan ke daftar aturan.
5. Konfigurasi aturan Anda:
 - a. Untuk Nama, masukkan nama untuk aturan Anda.
 - b. Untuk Untuk, pilih jenis objek yang berlaku aturan.
 - c. Untuk tempat, masukkan filter untuk diterapkan ke objek sebelum menerapkan aturan migrasi. The where klausa dievaluasi dengan menggunakan klausa like. Anda dapat memasukkan nama yang tepat untuk memilih satu objek, atau Anda dapat memasukkan pola untuk memilih beberapa objek.

Bidang yang tersedia untuk klausa where berbeda tergantung pada jenis objek. Misalnya, jika jenis objek skema hanya ada satu bidang yang tersedia, untuk nama skema.

- d. Untuk Tindakan, pilih jenis aturan migrasi yang ingin Anda buat.
- e. Tergantung pada jenis aturan, masukkan satu atau dua nilai tambahan. Misalnya, untuk mengganti nama objek, masukkan nama baru objek. Untuk mengganti awalan, masukkan awalan lama dan awalan baru.

Untuk char, varchar, nvarchar, dan tipe data string, Anda dapat mengubah panjang tipe data menggunakan operator perkalian. Misalnya, `%*4` nilai mengubah tipe `varchar(10)` data menjadi `varchar(40)`.

6. Setelah mengonfigurasi aturan migrasi, pilih Simpan untuk menyimpan aturan. Anda juga dapat memilih Batal untuk membatalkan perubahan.

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. Setelah selesai menambahkan, mengedit, dan menghapus aturan, pilih Simpan Semua untuk menyimpan semua perubahan.
8. Memilih Tutup untuk menutup Aturan transformasi kotak dialog.

Anda dapat menggunakan ikon toggle untuk menonaktifkan aturan migrasi tanpa menghapusnya. Anda dapat menggunakan ikon salin untuk menduplikasi aturan migrasi yang ada. Anda dapat menggunakan ikon pensil untuk mengedit aturan migrasi yang ada. Anda dapat menggunakan ikon

hapus untuk menghapus aturan migrasi yang ada. Untuk menyimpan perubahan apa pun yang Anda buat pada aturan migrasi, pilih Simpan Semua.

Mengekspor aturan migrasi

Jika digunakan AWS DMS untuk memigrasi data dari database sumber ke database target, Anda dapat memberikan informasi tentang aturan migrasi ke AWS DMS. Untuk informasi selengkapnya tentang tugas, lihat [Bekerja dengan tugas AWS Database Migration Service replikasi](#).

Untuk mengekspor aturan migrasi

1. Di bagian AWS Schema Conversion Tool, pilih Tampilan Pemetaan pada menu Tampilan.
2. Di Aturan migrasi, pilih aturan migrasi, lalu pilih Ubah aturan migrasi.
3. Pilih Ekspor skrip untuk AWS DMS.
4. Jelajahi lokasi tempat Anda ingin menyimpan skrip, lalu pilih Simpan. Aturan migrasi Anda disimpan sebagai skrip JSON yang dapat digunakan oleh AWS DMS.

Mengonversi skema Anda dengan menggunakan AWS SCT

Setelah Anda menghubungkan proyek Anda ke database sumber dan instans DB Amazon RDS target Anda, AWS Schema Conversion Tool proyek Anda menampilkan skema dari database sumber Anda di panel kiri. Skema disajikan dalam format pohon-view, dan setiap node pohon malas dimuat. Ketika Anda memilih node dalam tampilan pohon, AWS SCT meminta informasi skema dari database sumber Anda pada saat itu.

Anda dapat memilih item skema dari database sumber Anda dan kemudian mengonversi skema menjadi skema yang setara untuk mesin DB instans DB target Anda. Anda dapat memilih item skema apa pun dari database sumber Anda untuk dikonversi. Jika item skema yang Anda pilih tergantung pada item induk, maka AWS SCT juga menghasilkan skema untuk item induk. Misalnya, anggaplah Anda memilih tabel untuk dikonversi. Jika demikian, AWS SCT menghasilkan skema untuk tabel, dan database yang tabel di.

Mengubah skema

Untuk mengonversi skema dari database sumber Anda, pilih kotak centang untuk nama skema yang akan dikonversi. Selanjutnya, pilih skema ini dari panel kiri proyek Anda. AWS SCT menyoroti nama skema dengan warna biru. Buka menu konteks (klik kanan) untuk skema, dan pilih Convert schema, seperti yang ditunjukkan berikut.

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)>

Setelah Anda mengubah skema dari database sumber Anda, Anda dapat memilih item skema dari panel kiri proyek Anda dan melihat skema yang dikonversi di panel tengah proyek Anda. Panel bawah-tengah menampilkan properti dan perintah SQL untuk membuat skema dikonversi, seperti yang ditunjukkan berikut.

The screenshot shows the AWS Schema Conversion Tool interface. On the left, a tree view shows the server hierarchy: Servers > SQL Server - ec2-52-17-21-76.eu-west-1.cc > Databases [12] > LARGE_DB_SS > Schemas [2] > dbo > Tables [4] > Account. The main pane displays the SQL script for creating the table in the target Amazon RDS instance:

```

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, the cursor position is 0 and modified is true. The target Amazon RDS instance is identified as 'Target Amazon RDS for MySQL table: Account'. The right pane shows the converted SQL script for the target instance:

```

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,

```

Setelah Anda telah dikonversi skema Anda, Anda dapat menyimpan proyek Anda. Informasi skema dari database sumber Anda disimpan dengan proyek Anda. Fungsionalitas ini berarti Anda dapat bekerja secara offline tanpa terhubung ke database sumber Anda. AWS SCT terhubung ke database sumber Anda untuk memperbarui skema dalam proyek Anda jika Anda memilih Refresh from Database untuk database sumber Anda. Untuk informasi selengkapnya, lihat [Memperbarui dan menyegarkan skema yang telah dikonversi AWS SCT](#).

Anda dapat membuat laporan penilaian migrasi database dari item yang tidak dapat dikonversi secara otomatis. Laporan penilaian berguna untuk mengidentifikasi dan menyelesaikan item skema yang tidak dapat dikonversi secara otomatis. Untuk informasi selengkapnya, lihat [Membuat laporan penilaian migrasi dengan AWS SCT](#).

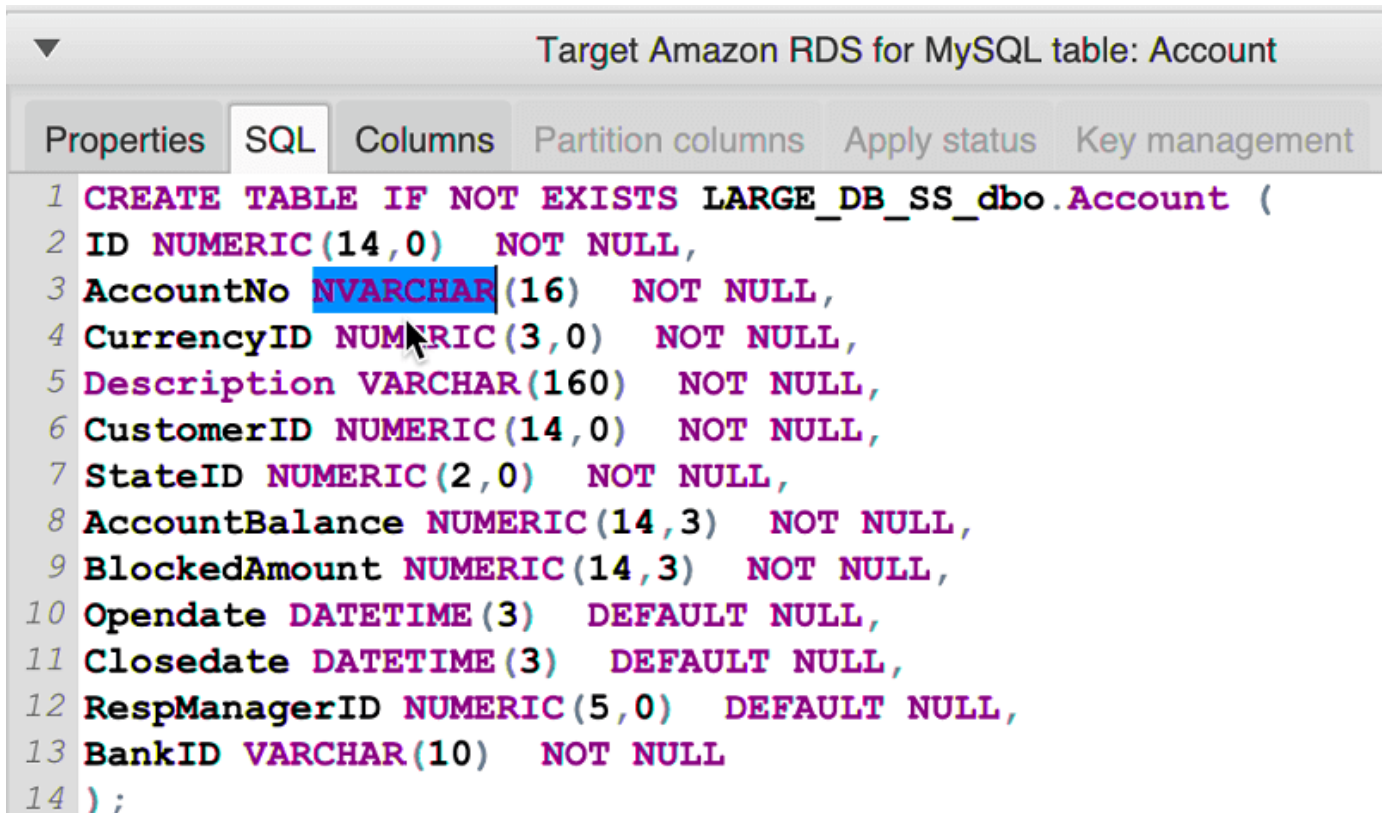
Ketika AWS SCT menghasilkan skema dikonversi, itu tidak segera menerapkannya ke contoh DB target. Sebaliknya, skema dikonversi disimpan secara lokal sampai Anda siap untuk menerapkannya ke instans DB target. Untuk informasi selengkapnya, lihat [Menerapkan skema yang telah dikonversi](#).

Mengedit skema yang dikonversi

Anda dapat mengedit skema dikonversi dan menyimpan perubahan sebagai bagian dari proyek Anda.

Untuk mengedit skema yang dikonversi

1. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih item skema yang ingin Anda edit skema yang dikonversi.
2. Di panel tengah bawah yang menampilkan skema yang dikonversi untuk item yang dipilih, pilih tab SQL.
3. Dalam teks yang ditampilkan untuk tab SQL, ubah skema sesuai kebutuhan. Skema secara otomatis disimpan dengan proyek Anda saat Anda memperbaruinya.



The screenshot shows the AWS Schema Conversion Tool interface. At the top, it says "Target Amazon RDS for MySQL table: Account". Below this, there are several tabs: "Properties", "SQL", "Columns", "Partition columns", "Apply status", and "Key management". The "SQL" tab is selected. The SQL code displayed is as follows:

```
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 );
```

Perubahan yang Anda buat untuk skema dikonversi disimpan dengan proyek Anda saat Anda melakukan pembaruan. Jika Anda baru saja mengonversi item skema dari database sumber Anda, dan Anda telah membuat pembaruan ke skema yang dikonversi sebelumnya untuk item tersebut, pembaruan yang ada akan digantikan oleh item skema yang baru dikonversi berdasarkan basis data sumber Anda.

Menghapus skema yang dikonversi

Sampai Anda menerapkan skema ke instans DB target Anda, AWS SCT hanya menyimpan skema yang dikonversi secara lokal di proyek Anda. Anda dapat menghapus skema yang direncanakan dari proyek Anda dengan memilih node tampilan pohon untuk instans DB Anda, dan kemudian memilih Refresh from Database. Karena tidak ada skema yang ditulis ke instans DB target Anda, penyegaran dari database menghapus elemen skema yang direncanakan dalam AWS SCT proyek Anda agar sesuai dengan apa yang ada di instans DB sumber Anda.

Menangani konversi manual AWS SCT

Laporan penilaian menyertakan daftar item yang tidak dapat dikonversi secara otomatis ke mesin database instans DB Amazon RDS target Anda. Untuk setiap item yang tidak dapat dikonversi, ada item tindakan di Item Tindakan tab.

Anda dapat menanggapi item tindakan dalam laporan penilaian dengan cara berikut:

- Memodifikasi skema database sumber Anda.
- Memodifikasi skema database target Anda.

Memodifikasi skema sumber

Untuk beberapa item, mungkin lebih mudah untuk memodifikasi skema database dalam database sumber Anda ke skema yang dapat dikonversi secara otomatis. Pertama, verifikasi bahwa perubahan baru kompatibel dengan arsitektur aplikasi Anda, lalu perbarui skema di database sumber Anda. Terakhir, segarkan proyek Anda dengan informasi skema yang diperbarui. Anda kemudian dapat mengonversi skema yang diperbarui, dan membuat laporan penilaian migrasi database baru. Item tindakan tidak lagi muncul untuk item yang berubah dalam skema sumber.

Keuntungan dari proses ini adalah skema yang diperbarui selalu tersedia saat Anda menyegarkan dari database sumber Anda.

Memodifikasi skema target

Untuk beberapa item, mungkin lebih mudah untuk menerapkan skema yang dikonversi ke database target Anda, dan kemudian menambahkan item skema setara secara manual ke database target Anda untuk item yang tidak dapat dikonversi secara otomatis. Anda dapat menulis semua skema

yang dapat dikonversi secara otomatis ke instans DB target Anda dengan menerapkan skema. Untuk informasi selengkapnya, lihat [Menyimpan dan menerapkan skema yang telah dikonversi AWS SCT](#).

Skema yang ditulis ke instans DB target Anda tidak berisi item yang tidak dapat dikonversi secara otomatis. Setelah menerapkan skema ke instans DB target Anda, Anda kemudian dapat secara manual membuat skema dalam instans DB target Anda yang setara dengan yang ada di database sumber. Item tindakan dalam laporan penilaian migrasi database berisi saran tentang cara membuat skema yang setara.

Warning

Jika Anda membuat skema secara manual dalam instans DB target Anda, simpan salinan pekerjaan manual apa pun yang Anda lakukan. Jika Anda menerapkan skema yang dikonversi dari proyek Anda ke instans DB target Anda lagi, itu akan menimpa pekerjaan manual yang telah Anda lakukan.

Dalam beberapa kasus, Anda tidak dapat membuat skema setara dalam instans DB target Anda. Anda mungkin perlu merancang ulang sebagian aplikasi dan database Anda untuk menggunakan fungsionalitas yang tersedia dari mesin DB untuk instans DB target Anda. Dalam kasus lain, Anda dapat mengabaikan skema yang tidak dapat dikonversi secara otomatis.

Memperbarui dan menyegarkan skema yang telah dikonversi AWS SCT

Anda dapat memperbarui skema sumber dan skema target dalam proyek Anda AWS Schema Conversion Tool.

- **Sumber** - Jika Anda memperbarui skema untuk database sumber Anda, AWS SCT gantikan skema dalam proyek Anda dengan skema terbaru dari database sumber Anda. Dengan menggunakan fungsionalitas ini, Anda dapat memperbarui proyek Anda jika perubahan telah dilakukan pada skema database sumber Anda.
- **Target** — Jika Anda memperbarui skema untuk instans DB Amazon RDS target Anda, AWS SCT gantikan skema dalam proyek Anda dengan skema terbaru dari instans DB target Anda. Jika Anda belum menerapkan skema apa pun ke instans DB target Anda, AWS SCT hapus skema yang dikonversi dari proyek Anda. Anda kemudian dapat mengkonversi skema dari database sumber Anda untuk contoh DB target bersih.

Anda memperbarui skema dalam AWS SCT proyek Anda dengan memilih Refresh from Database.

Note

Ketika Anda me-refresh skema Anda, AWS SCT memuat metadata hanya seperti yang diperlukan. Untuk memuat sepenuhnya semua skema database Anda, buka menu konteks (klik kanan) untuk skema Anda, dan pilih Load schema. Misalnya, Anda dapat menggunakan opsi ini untuk memuat metadata untuk database Anda sekaligus, dan kemudian bekerja secara offline.

Menyimpan dan menerapkan skema yang telah dikonversi AWS SCT

Ketika AWS Schema Conversion Tool menghasilkan skema dikonversi (seperti yang ditunjukkan dalam [Mengonversi skema Anda dengan menggunakan AWS SCT](#)), itu tidak segera menerapkan skema dikonversi ke contoh DB target. Sebaliknya, skema dikonversi disimpan secara lokal dalam proyek Anda sampai Anda siap untuk menerapkannya ke instans DB target. Dengan menggunakan fungsionalitas ini, Anda dapat bekerja dengan item skema yang tidak dapat dikonversi secara otomatis ke mesin DB target Anda. Untuk informasi selengkapnya tentang item yang tidak dapat dikonversi secara otomatis, lihat [Membuat laporan penilaian migrasi dengan AWS SCT](#).

Anda opsional dapat memiliki alat menyimpan skema Anda dikonversi ke file sebagai script SQL sebelum menerapkan skema untuk contoh DB target Anda. Anda juga dapat memiliki alat menerapkan skema dikonversi langsung ke instans DB target Anda.

Menyimpan skema yang telah dikonversi ke file

Anda dapat menyimpan skema yang telah dikonversi sebagai skrip SQL dalam file teks. Dengan menggunakan pendekatan ini, Anda dapat memodifikasi skrip SQL yang dihasilkan dari AWS SCT ke item alamat yang alat tidak dapat dikonversi secara otomatis. Anda kemudian dapat menjalankan skrip yang diperbarui pada instans DB target Anda untuk menerapkan skema yang telah dikonversi ke database target Anda.

Untuk menyimpan skema Anda dikonversi sebagai skrip SQL

1. Pilih skema Anda dan buka menu konteks (klik kanan).
2. Pilih Simpan sebagai SQL.

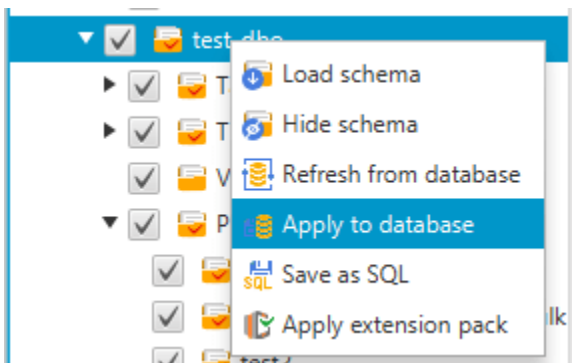
3. Masukkan nama file dan pilih Simpan.
4. Simpan skema Anda yang telah dikonversi menggunakan salah satu opsi berikut:
 - Berkas tunggal
 - File tunggal per tahap
 - File tunggal per pernyataan

Untuk memilih format skrip SQL

1. Pada menu Pengaturan, pilih Pengaturan proyek.
2. Pilih Simpan skrip.
3. Untuk Vendor, pilih platform database.
4. Untuk Simpan skrip SQL, pilih bagaimana Anda ingin menyimpan skrip skema database Anda.
5. Pilih OK untuk menyimpan pengaturan.

Menerapkan skema yang telah dikonversi

Saat Anda siap menerapkan skema yang telah dikonversi ke instans DB Amazon RDS target Anda, pilih elemen skema dari panel kanan proyek Anda. Buka konteks (klik kanan) menu untuk elemen skema, dan kemudian pilih Terapkan ke database, seperti yang ditunjukkan berikut.



Skema paket ekstensi

Pertama kali Anda menerapkan skema yang dikonversi ke instans DB target Anda, AWS SCT tambahkan skema tambahan ke instans DB target Anda. Skema ini mengimplementasikan fungsi sistem dari database sumber yang diperlukan saat menulis skema yang dikonversi ke instans DB target Anda. Skema ini disebut skema paket ekstensi.

Jangan memodifikasi skema paket ekstensi, atau Anda mungkin menemukan hasil yang tidak terduga dalam skema yang dikonversi yang ditulis ke instans DB target Anda. Ketika skema Anda sepenuhnya dimigrasi ke instans DB target Anda, dan Anda tidak perlu lagi AWS SCT, Anda dapat menghapus skema paket ekstensi.

Skema paket ekstensi diberi nama sesuai dengan database sumber Anda sebagai berikut:

- IBM Db2 LUW: `aws_db2_ext`
- Server Microsoft SQL: `aws_sqlserver_ext`
- MySQL: `aws_mysql_ext`
- Oracle: `aws_oracle_ext`
- PostgreSQL: `aws_postgresql_ext`
- SAP ASE: `aws_sapase_ext`

Untuk informasi selengkapnya, lihat [Menggunakan AWS Lambda fungsi dari paket AWS SCT ekstensi](#).

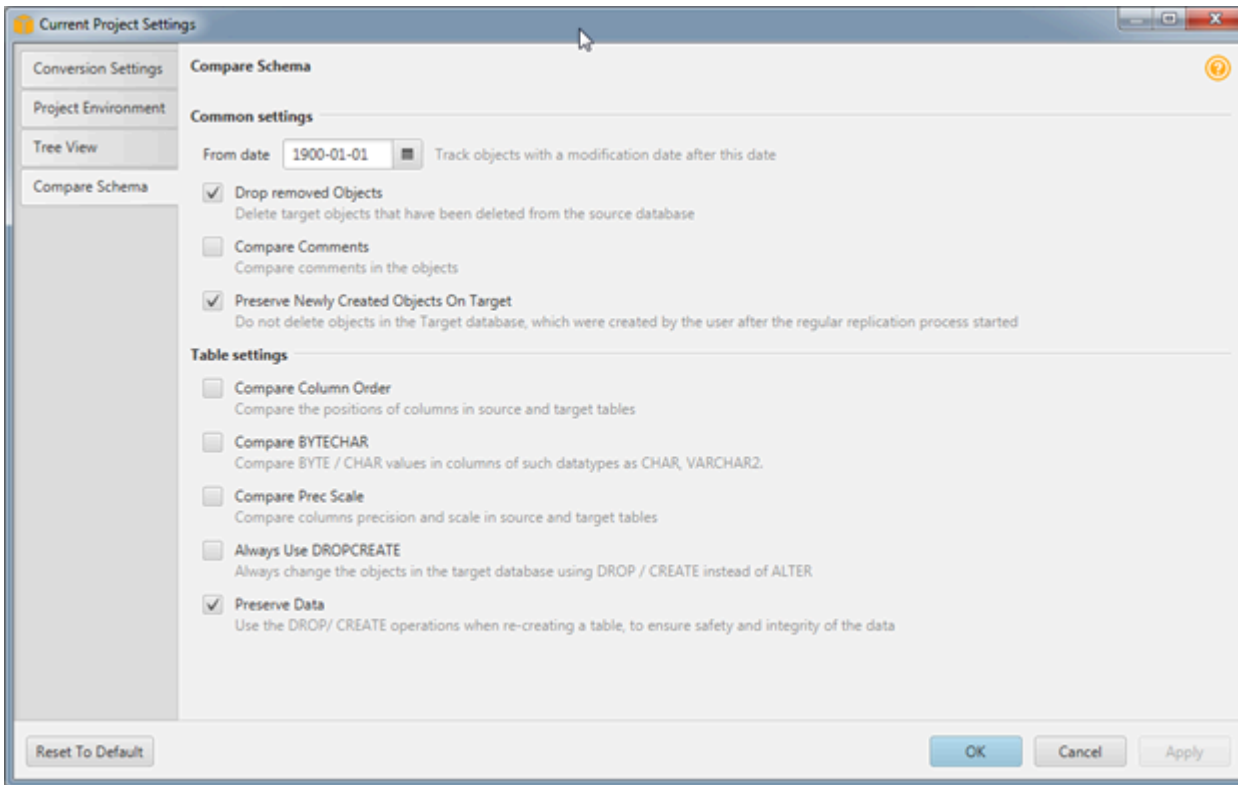
Membandingkan skema database

Jika Anda membuat perubahan pada skema sumber atau target Anda setelah Anda bermigrasi, Anda dapat membandingkan dua skema database menggunakan AWS SCT. Anda dapat membandingkan skema untuk versi yang sama atau lebih awal dari skema sumber.

Perbandingan skema berikut didukung:

- Oracle ke Oracle, versi 12.1.0.2.0, 11.1.0.7.0, 11.2.0.1.0, 10
- SQL Server ke SQL Server, versi 2016, 2014, 2012, 2008 RD2, 2008
- PostgreSQL ke PostgreSQL dan Edisi yang kompatibel dengan Aurora PostgreSQL, versi 9.6, 9.5.9, 9.5.4
- MySQL ke MySQL, versi 5.6.36, 5.7.17, 5.5

Anda menentukan pengaturan untuk perbandingan skema di Bandingkan Skema tab dari halaman Pengaturan Proyek.

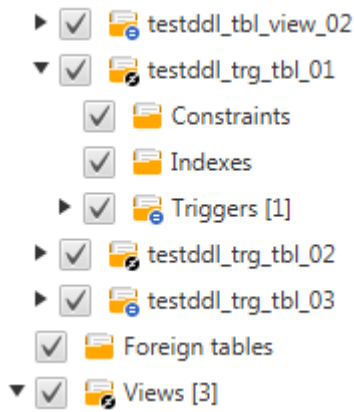


Untuk membandingkan skema, Anda memilih skema, dan AWS SCT menunjukkan objek yang berbeda antara dua skema dan objek yang tidak.

Untuk membandingkan dua skema

1. Buka AWS SCT proyek yang ada, atau buat proyek dan hubungkan ke titik akhir sumber dan target.
2. Pilih skema yang ingin Anda bandingkan.
3. Buka menu konteks (klik kanan) dan pilih Bandingkan Skema.

AWS SCT menunjukkan objek yang berbeda antara dua skema dengan menambahkan lingkaran hitam ke ikon objek.



Anda dapat menerapkan hasil perbandingan skema ke satu objek, ke satu kategori objek, atau ke seluruh skema. Pilih kotak di samping kategori, objek, atau skema yang ingin Anda terapkan hasilnya.

Menemukan objek yang ditransformasikan terkait

Setelah konversi skema, dalam beberapa kasus AWS SCT mungkin telah menciptakan beberapa objek untuk satu objek skema pada database sumber. Misalnya, ketika melakukan konversi Oracle ke PostgreSQL, AWS SCT mengambil setiap pemicu Oracle dan mengubahnya menjadi pemicu dan memicu fungsi pada target PostgreSQL. Juga, ketika AWS SCT mengkonversi fungsi paket Oracle atau prosedur untuk PostgreSQL, menciptakan fungsi setara dan fungsi INIT yang harus dijalankan sebagai blok init sebelum prosedur atau fungsi dapat dijalankan.

Prosedur berikut memungkinkan Anda melihat semua objek terkait yang dibuat setelah konversi skema.

Untuk melihat objek terkait yang dibuat selama konversi skema

1. Setelah konversi skema, pilih objek yang dikonversi dalam tampilan pohon target.
2. Pilih tab Objek Konversi Terkait.
3. Lihat daftar objek target terkait.

Mengonversi skema gudang data ke Amazon Redshift menggunakan AWS SCT

AWS Schema Conversion Tool(AWS SCT) mengotomatiskan sebagian besar proses konversi skema gudang data Anda ke skema database Amazon Redshift. Karena sumber dan target mesin database dapat memiliki banyak fitur yang berbeda dan kemampuan, AWS SCT mencoba untuk membuat skema setara dalam database target Anda sedapat mungkin. Jika tidak ada konversi langsung yang memungkinkan, AWS SCT berikan laporan penilaian dengan daftar tindakan yang mungkin untuk Anda ambil. Dengan menggunakan AWS SCT, Anda dapat mengelola kunci, memetakan jenis data dan objek, dan membuat konversi manual.

AWS SCT dapat mengonversi skema gudang data berikut ke Amazon Redshift.

- Amazon Redshift
- Azure Synapse Analytics (versi 10)
- BigQuery
- Database Greenplum (versi 4.3)
- Microsoft SQL Server (versi 2008 dan yang lebih tinggi)
- Netezza (versi 7.0.3 dan lebih tinggi)
- Oracle (versi 10.2 dan lebih tinggi)
- Kepingan salju (versi 3)
- Teradata (versi 13 dan lebih tinggi)
- Vertica (versi 7.2 dan lebih tinggi)

Untuk informasi tentang mengonversi skema database pemrosesan transaksi online (OLTP), lihat.

[Mengonversi skema database menggunakan AWS SCT](#)

Untuk mengonversi skema gudang data, lakukan langkah-langkah berikut:

1. Tentukan strategi dan aturan pengoptimalan, dan tentukan aturan migrasi yang AWS SCT ingin Anda gunakan. Anda dapat mengatur aturan yang mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek.

Anda dapat menentukan aturan pengoptimalan dan migrasi di Pengaturan. Untuk informasi selengkapnya tentang strategi pengoptimalan, [Memilih strategi dan aturan pengoptimalan untuk](#)

- [digunakan AWS SCT](#) lihat. untuk informasi selengkapnya tentang aturan migrasi, lihat [Membuat aturan migrasi di AWS SCT](#)
2. Berikan statistik pada gudang data sumber Anda sehingga AWS SCT dapat mengoptimalkan bagaimana gudang data Anda dikonversi. Anda dapat mengumpulkan statistik langsung dari database, atau mengunggah file statistik yang ada. Untuk informasi lebih lanjut tentang menyediakan statistik gudang data, lihat [Mengumpulkan atau mengunggah statistik untuk AWS SCT](#).
 3. Buat laporan penilaian migrasi database yang merinci elemen skema yang tidak dapat dikonversi secara otomatis. Anda dapat menggunakan laporan ini untuk mengidentifikasi di mana Anda perlu membuat skema secara manual dalam database target Anda yang kompatibel dengan database sumber Anda. Untuk informasi lebih lanjut tentang laporan penilaian, lihat [Membuat laporan penilaian migrasi dengan AWS SCT](#).
 4. Mengkonversi skema. AWS SCT membuat versi lokal dari skema yang dikonversi untuk Anda tinjau, tetapi tidak menerapkannya ke database target Anda sampai Anda siap. Untuk informasi selengkapnya tentang mengonversi, lihat [Mengonversi skema Anda menggunakan AWS SCT](#)
 5. Setelah mengonversi skema, Anda dapat mengelola dan mengedit kunci Anda. Manajemen kunci adalah jantung dari konversi gudang data. Untuk informasi selengkapnya tentang mengelola kunci, lihat [Mengelola dan menyesuaikan kunci di AWS SCT](#).
 6. Jika Anda memiliki elemen skema yang tidak dapat dikonversi secara otomatis, Anda memiliki dua pilihan: memperbarui skema sumber dan kemudian mengkonversi lagi, atau membuat elemen skema setara dalam database target Anda. Untuk informasi lebih lanjut tentang mengkonversi elemen skema secara manual, lihat. [Menangani konversi manual di AWS SCT](#) Untuk informasi selengkapnya tentang memperbarui skema sumber Anda, lihat [Memperbarui dan menyegarkan skema yang telah dikonversi AWS SCT](#).
 7. Ketika Anda siap, Anda dapat menerapkan skema dikonversi ke database target Anda. Untuk informasi lebih lanjut tentang menyimpan dan menerapkan skema dikonversi, lihat [Menyimpan dan menerapkan skema yang telah dikonversi AWS SCT](#).

Izin untuk Amazon Redshift sebagai target

Izin yang diperlukan untuk Amazon Redshift sebagai target dicantumkan sebagai berikut:

- CREATE ON DATABASE - memungkinkan untuk membuat skema baru dalam database.
- CREATE ON SCHEMA - memungkinkan untuk membuat objek dalam skema database.

- PENGGUNAAN HIBAH PADA BAHASA - memungkinkan untuk membuat fungsi dan prosedur baru dalam database.
- GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog — memberi pengguna informasi sistem tentang klaster Amazon Redshift.
- GRANT SELECT ON pg_class_info - menyediakan pengguna dengan informasi tentang gaya distribusi tabel.

Anda dapat menggunakan contoh kode berikut untuk membuat pengguna database dan memberikan izin.

```
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;
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *db_name* dengan nama database Amazon Redshift target Anda. Selanjutnya, ganti *schema_name* dengan nama skema Amazon Redshift Anda. Ulangi GRANT CREATE ON SCHEMA operasi untuk setiap skema target di mana Anda akan menerapkan kode yang dikonversi atau memigrasi data. Terakhir, ganti kata *sandi Anda dengan kata sandi* yang aman.

Anda dapat menerapkan paket ekstensi pada database Amazon Redshift target Anda. Paket ekstensi adalah modul add-on yang mengemulasi fungsi database sumber yang diperlukan saat mengonversi objek ke Amazon Redshift. Untuk informasi selengkapnya, lihat [Menggunakan paket AWS SCT ekstensi](#).

Untuk operasi ini, AWS SCT memerlukan izin untuk mengakses bucket Amazon S3 atas nama Anda. Untuk memberikan izin ini, buat pengguna AWS Identity and Access Management (IAM) dengan kebijakan berikut.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Action": "s3:ListBucket",  
      "Resource": "arn:aws:s3:::bucket_name",  
      "Effect": "Allow",  
      "Principal": "AWS:*",  
      "Condition": {"StringEquals": {"aws:PrincipalTag": "IAMRoleOnlyUseForS3Access"}}  
    }  
  ]  
}
```

```
{
  "Effect": "Allow",
  "Action": [
    "s3:DeleteObject",
    "s3:GetObject",
    "s3:ListBucket",
    "s3:PutObject"
  ],
  "Resource": [
    "arn:aws:s3:::aws-sct-*"
  ]
},
{
  "Effect": "Allow",
  "Action": [
    "s3:ListAllMyBuckets"
  ],
  "Resource": ""
}
]
```

Memilih strategi dan aturan pengoptimalan untuk digunakan AWS SCT

Untuk mengoptimalkan bagaimana AWS Schema Conversion Tool mengkonversi skema gudang data Anda, Anda dapat memilih strategi dan aturan yang Anda inginkan alat untuk digunakan. Setelah mengonversi skema, dan meninjau kunci yang disarankan, Anda dapat menyesuaikan aturan atau mengubah strategi Anda untuk mendapatkan hasil yang Anda inginkan.

Untuk memilih strategi dan aturan pengoptimalan Anda

1. Pilih Pengaturan, lalu pilih Pengaturan Proyek. Kotak dialog pengaturan proyek saat ini muncul.
2. Di panel kiri, pilih Strategi Pengoptimalan. Strategi pengoptimalan muncul di panel kanan dengan default yang dipilih.
3. Untuk Sektor Strategi, pilih strategi pengoptimalan yang ingin Anda gunakan. Anda dapat memilih dari opsi berikut:
 - Gunakan metadata, abaikan informasi statistik - Dalam strategi ini, hanya informasi dari metadata yang digunakan untuk keputusan pengoptimalan. Misalnya, jika ada lebih dari satu

indeks pada tabel sumber, urutan urutan database sumber digunakan, dan indeks pertama menjadi kunci distribusi.

- Abaikan metadata, gunakan informasi statistik - Dalam strategi ini, keputusan pengoptimalan hanya berasal dari informasi statistik saja. Strategi ini hanya berlaku untuk tabel dan kolom yang statistiknya disediakan. Untuk informasi selengkapnya, lihat [Mengumpulkan atau mengunggah statistik untuk AWS SCT](#).
 - Gunakan metadata dan gunakan informasi statistik - Dalam strategi ini, metadata dan statistik digunakan untuk keputusan pengoptimalan.
4. Setelah Anda memilih strategi pengoptimalan, Anda dapat memilih aturan yang ingin Anda gunakan. Anda dapat memilih dari opsi berikut:
- Pilih Distribution Key dan Sort Keys menggunakan metadata
 - Pilih tabel fakta dan dimensi yang sesuai untuk pemeriksaan
 - Menganalisis kardinalitas kolom indeks
 - Temukan tabel dan kolom yang paling sering digunakan dari tabel log kueri

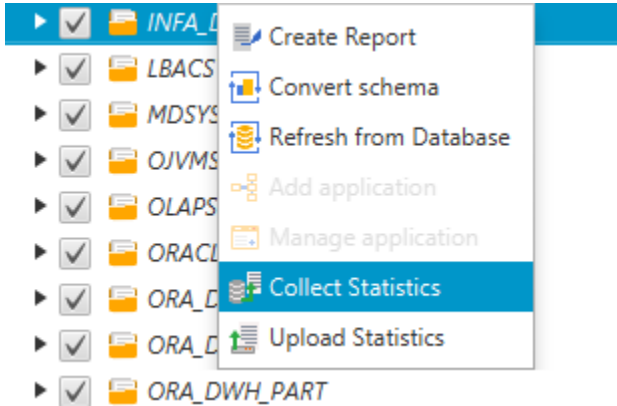
Untuk setiap aturan, Anda dapat memasukkan bobot untuk kunci sortir dan bobot untuk kunci distribusi. AWS SCT menggunakan bobot yang Anda pilih saat mengubah skema Anda. Kemudian, ketika Anda meninjau kunci yang disarankan, jika Anda tidak puas dengan hasilnya, Anda dapat kembali ke sini dan mengubah pengaturan Anda. Untuk informasi selengkapnya, lihat [Mengelola dan menyesuaikan kunci di AWS SCT](#).

Mengumpulkan atau mengunggah statistik untuk AWS SCT

Untuk mengoptimalkan bagaimana AWS Schema Conversion Tool mengkonversi skema gudang data Anda, Anda dapat memberikan statistik dari database sumber Anda bahwa alat dapat menggunakan. Anda dapat mengumpulkan statistik langsung dari database, atau mengunggah file statistik yang ada.

Untuk menyediakan dan meninjau statistik

1. Buka proyek Anda dan sambungkan ke database sumber Anda.
2. Pilih objek skema dari panel kiri proyek Anda, dan buka menu konteks (klik kanan) untuk objek. Pilih Kumpulkan Statistik atau Unggah Statistik seperti yang ditunjukkan berikut.



3. Pilih objek skema dari panel kiri proyek Anda, dan kemudian pilih tab Statistik. Anda dapat meninjau statistik untuk objek.

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

Kemudian, ketika Anda meninjau kunci yang disarankan, jika Anda tidak puas dengan hasilnya, Anda dapat mengumpulkan statistik tambahan dan mengulangi prosedur ini. Untuk informasi selengkapnya, lihat [Mengelola dan menyesuaikan kunci di AWS SCT](#).

Membuat aturan migrasi di AWS SCT

Sebelum mengonversi skema AWS SCT, Anda dapat mengatur aturan migrasi. Aturan migrasi dapat melakukan hal-hal seperti mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek. Misalnya, Anda memiliki satu set tabel dalam skema sumber Anda bernama `test_TABLE_NAME`. Anda dapat mengatur aturan yang mengubah awalan `test_` ke awalan `demo_` dalam skema target.

Note

Anda dapat membuat aturan migrasi hanya untuk mesin database sumber dan target yang berbeda.

Anda dapat membuat aturan migrasi yang melakukan tugas-tugas berikut:

- Menambahkan, menghapus, atau mengganti awalan
- Menambahkan, menghapus, atau mengganti akhiran
- Ubah kolasi kolom
- Ubah tipe data
- Mengubah panjang `char`, `varchar`, `nvarchar`, dan tipe `string` data
- Pindahkan objek
- Ubah nama objek

Anda dapat membuat aturan migrasi untuk objek berikut:

- Basis Data
- Skema
- Tabel
- Kolom

Membuat aturan migrasi

Anda dapat membuat aturan migrasi dan menyimpan aturan sebagai bagian dari proyek Anda. Dengan proyek Anda terbuka, gunakan prosedur berikut untuk membuat aturan migrasi.

Membuat aturan migrasi

1. Pada menu View, pilih Tampilan pemetaan.
2. Di Pemetaan server, pilih sepasang server sumber dan target.
3. Pilih Aturan migrasi baru. Kotak dialog Transformation rules muncul.
4. Pilih Tambahkan aturan baru. Baris baru ditambahkan ke daftar aturan.
5. Konfigurasi aturan Anda:
 - a. Untuk Nama, masukkan nama untuk aturan Anda.
 - b. Untuk Untuk, pilih jenis objek yang berlaku aturan.
 - c. Untuk tempat, masukkan filter untuk diterapkan ke objek sebelum menerapkan aturan migrasi. The where klausa dievaluasi dengan menggunakan klausa like. Anda dapat memasukkan nama yang tepat untuk memilih satu objek, atau Anda dapat memasukkan pola untuk memilih beberapa objek.

Bidang yang tersedia untuk klausa where berbeda tergantung pada jenis objek. Misalnya, jika tipe objek adalah skema hanya ada satu bidang yang tersedia, untuk nama skema.
 - d. Untuk Tindakan, pilih jenis aturan migrasi yang ingin Anda buat.
 - e. Tergantung pada jenis aturan, masukkan satu atau dua nilai tambahan. Misalnya, untuk mengganti nama objek, masukkan nama baru objek. Untuk mengganti awalan, masukkan awalan lama dan awalan baru.
6. Setelah mengonfigurasi aturan migrasi, pilih Simpan untuk menyimpan aturan. Anda juga dapat memilih Batal untuk membatalkan perubahan.

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. Setelah selesai menambahkan, mengedit, dan menghapus aturan, pilih Simpan Semua untuk menyimpan semua perubahan.
8. Memilih Tutup untuk menutup Aturan transformasi kotak dialog.

Anda dapat menggunakan ikon toggle untuk menonaktifkan aturan migrasi tanpa menghapusnya. Anda dapat menggunakan ikon salin untuk menduplikasi aturan migrasi yang ada. Anda dapat menggunakan ikon pensil untuk mengedit aturan migrasi yang ada. Anda dapat menggunakan ikon hapus untuk menghapus aturan migrasi yang ada. Untuk menyimpan perubahan apa pun yang Anda buat pada aturan migrasi, pilih Simpan Semua.

Mengekspor aturan migrasi

Jika Anda menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasi data dari database sumber ke database target, Anda dapat memberikan informasi tentang aturan migrasi ke AWS DMS. Untuk informasi selengkapnya tentang tugas, lihat [Bekerja dengan tugas AWS Database Migration Service replikasi](#).

Untuk mengekspor aturan migrasi

1. Di bagian AWS Schema Conversion Tool, pilih Tampilan Pemetaan pada menu Tampilan.
2. Di Aturan migrasi, pilih aturan migrasi, lalu pilih Ubah aturan migrasi.
3. Pilih Ekspor skrip untuk AWS DMS.

4. Jelajahi lokasi tempat Anda ingin menyimpan skrip, lalu pilih Simpan. Aturan migrasi Anda disimpan sebagai skrip JSON yang dapat digunakan oleh AWS DMS.

Mengonversi skema Anda menggunakan AWS SCT

Setelah Anda menghubungkan proyek Anda ke database sumber dan database target Anda, AWS Schema Conversion Tool proyek Anda menampilkan skema dari database sumber Anda di panel kiri. Skema disajikan dalam format pohon-view, dan setiap simpul pohon malas dimuat. Ketika Anda memilih node dalam tampilan pohon, AWS SCT meminta informasi skema dari database sumber Anda pada saat itu.

Anda dapat memilih item skema dari database sumber Anda dan kemudian mengubah skema ke skema yang setara untuk mesin database database target Anda. Anda dapat memilih item skema apa pun dari database sumber Anda untuk dikonversi. Jika item skema yang Anda pilih tergantung pada item induk, maka AWS SCT juga menghasilkan skema untuk item induk. Misalnya, jika Anda memilih kolom dari tabel untuk mengkonversi, kemudian AWS SCT menghasilkan skema untuk kolom, tabel yang kolom di, dan database yang tabel di.

Mengubah skema

Untuk mengonversi skema dari database sumber Anda, pilih kotak centang untuk nama skema yang akan dikonversi. Selanjutnya, pilih skema ini dari panel kiri proyek Anda. AWS SCT menyoroti nama skema dengan warna biru. Buka menu konteks (klik kanan) untuk skema, dan pilih Convert schema, seperti yang ditunjukkan berikut.

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)>

Setelah Anda mengubah skema dari database sumber Anda, Anda dapat memilih item skema dari panel kiri proyek Anda dan melihat skema yang dikonversi di panel tengah proyek Anda. Panel bawah-tengah menampilkan properti dan perintah SQL untuk membuat skema dikonversi, seperti yang ditunjukkan berikut.

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the server structure: SQL Server - ec2-52-17-21-76.eu-west-1.cc > Databases [12] > AdventureWorks2012_CS > alfresco > GOLD_TEST_SS_PG > LARGE_DB_SS > Schemas [2] > dbo > Tables [4] > Account. The main area is split into two panes. The top pane shows the source SQL script for the 'Account' table in the 'dbo' schema, including columns like ID, AccountNo, CurrencyID, Description, CustomerID, StateID, AccountBalance, BlockedAmount, Operdate, Closedate, RespManagerID, and BankID. The bottom pane shows the converted MySQL script for the 'Account' table in the 'LARGE_DB_SS_dbo' schema, with columns converted to NUMERIC, VARCHAR, DATETIME, and DEFAULT NULL. The cursor position is 0 and modified is true.

Setelah Anda telah dikonversi skema Anda, Anda dapat menyimpan proyek Anda. Informasi skema dari database sumber Anda disimpan dengan proyek Anda. Fungsionalitas ini berarti Anda dapat bekerja secara offline tanpa terhubung ke database sumber Anda. AWS SCT terhubung ke database sumber Anda untuk memperbarui skema dalam proyek Anda jika Anda memilih Refresh from Database untuk database sumber Anda. Untuk informasi selengkapnya, lihat [Memperbarui dan menyegarkan skema yang telah dikonversi AWS SCT](#).

Anda dapat membuat laporan penilaian migrasi database dari item yang tidak dapat dikonversi secara otomatis. Laporan penilaian berguna untuk mengidentifikasi dan menyelesaikan item skema yang tidak dapat dikonversi secara otomatis. Untuk informasi selengkapnya, lihat [Membuat laporan penilaian migrasi dengan AWS SCT](#).

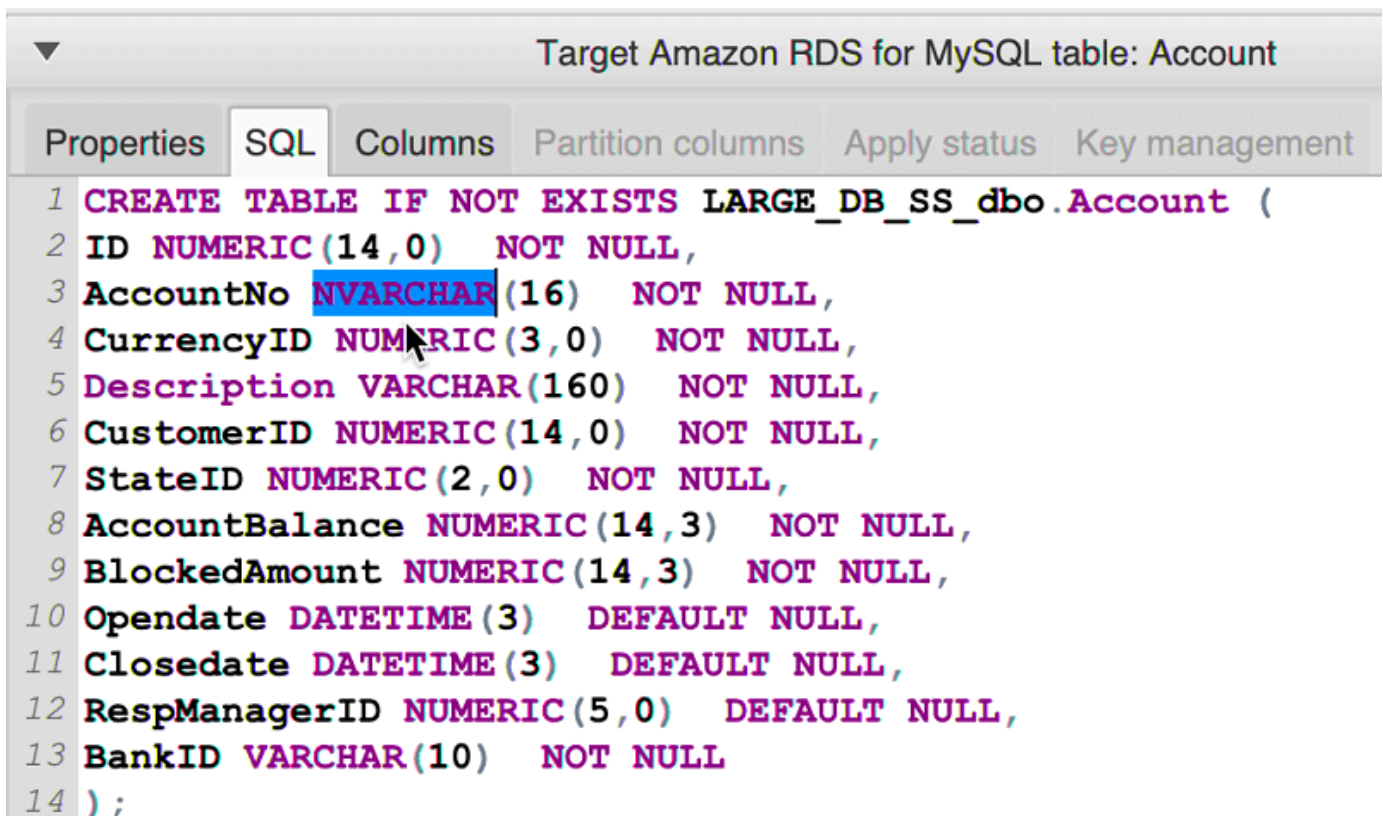
Ketika AWS SCT menghasilkan skema dikonversi, itu tidak segera menerapkannya ke database target. Sebaliknya, skema dikonversi disimpan secara lokal sampai Anda siap untuk menerapkannya ke database target. Untuk informasi selengkapnya, lihat [Menerapkan skema yang telah dikonversi](#).

Mengedit skema yang dikonversi

Anda dapat mengedit skema dikonversi dan menyimpan perubahan sebagai bagian dari proyek Anda.

Untuk mengedit skema yang dikonversi

1. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih item skema yang ingin Anda edit skema yang dikonversi.
2. Di panel tengah bawah yang menampilkan skema yang dikonversi untuk item yang dipilih, pilih tab SQL.
3. Dalam teks yang ditampilkan untuk tab SQL, ubah skema sesuai kebutuhan. Skema secara otomatis disimpan dengan proyek Anda saat Anda memperbaruinya.



The screenshot shows the AWS Schema Conversion Tool interface. At the top, it says "Target Amazon RDS for MySQL table: Account". Below this, there are several tabs: "Properties", "SQL", "Columns", "Partition columns", "Apply status", and "Key management". 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 );
```

Perubahan yang Anda buat untuk skema dikonversi disimpan dengan proyek Anda saat Anda melakukan pembaruan. Jika Anda baru saja mengonversi item skema dari database sumber Anda, dan Anda telah membuat pembaruan ke skema yang dikonversi sebelumnya untuk item tersebut, pembaruan yang ada akan digantikan oleh item skema yang baru dikonversi berdasarkan basis data sumber Anda.

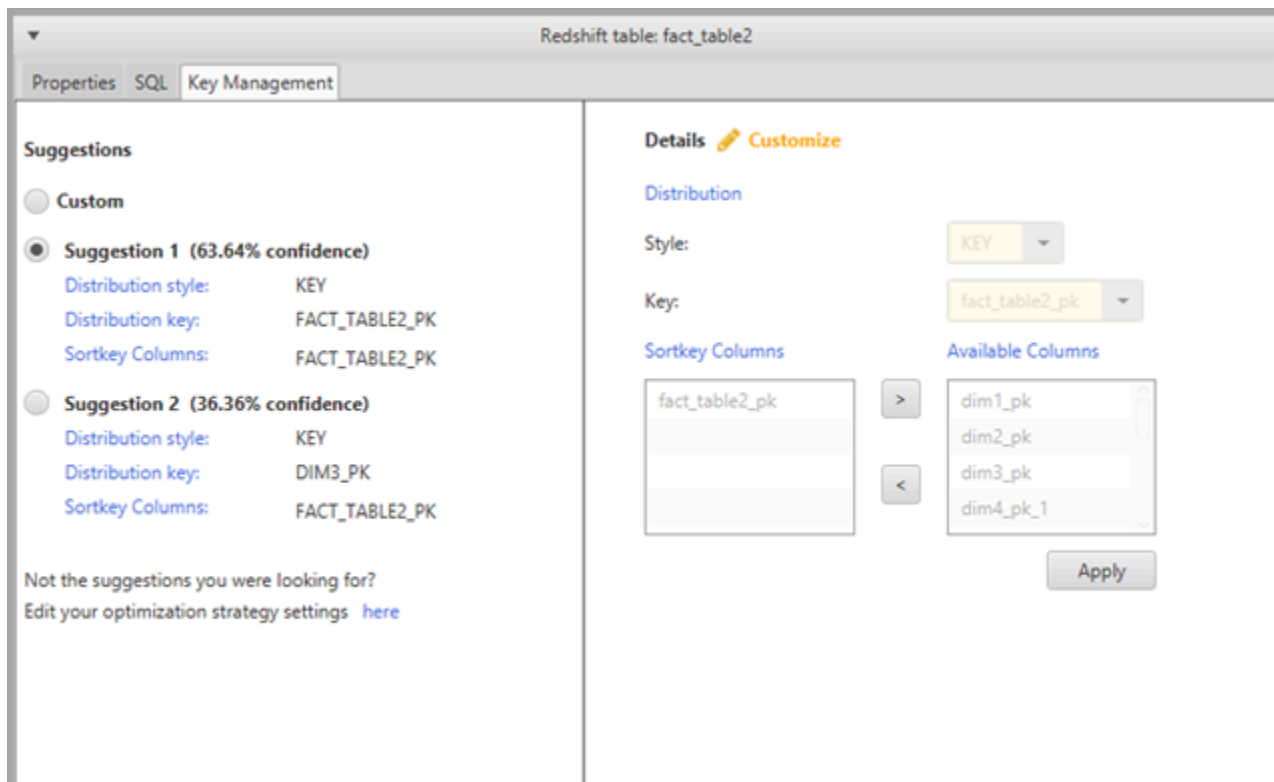
Menghapus skema yang dikonversi

Sampai Anda menerapkan skema ke database target Anda, AWS SCT hanya menyimpan skema dikonversi secara lokal dalam proyek Anda. Anda dapat menghapus skema yang direncanakan dari proyek Anda dengan memilih node tampilan pohon untuk database target Anda, dan kemudian memilih Refresh dari Database. Karena tidak ada skema yang ditulis ke database target Anda, refresh dari database menghapus elemen skema yang direncanakan dalam AWS SCT proyek Anda agar sesuai dengan apa yang ada di database target Anda.

Mengelola dan menyesuaikan kunci di AWS SCT

Setelah Anda mengkonversi skema Anda dengan AWS Schema Conversion Tool, Anda dapat mengelola dan mengedit kunci Anda. Manajemen kunci adalah jantung dari konversi gudang data.

Untuk mengelola kunci, pilih tabel di database target Anda, dan kemudian pilih tab Manajemen Kunci seperti yang ditunjukkan berikut.



Panel kiri berisi saran utama, dan menyertakan peringkat kepercayaan untuk setiap saran. Anda dapat memilih salah satu saran, atau Anda dapat menyesuaikan kunci dengan mengeditnya di panel kanan.

Jika pilihan untuk kunci tidak terlihat seperti yang Anda harapkan, Anda dapat mengedit strategi pengoptimalan Anda, dan kemudian mencoba kembali konversi. Untuk informasi selengkapnya, lihat [Memilih strategi dan aturan pengoptimalan untuk digunakan AWS SCT](#).

Topik terkait

- [Pilih kunci sortir terbaik](#)
- [Pilih gaya distribusi terbaik](#)

Membuat dan menggunakan laporan penilaian di AWS SCT

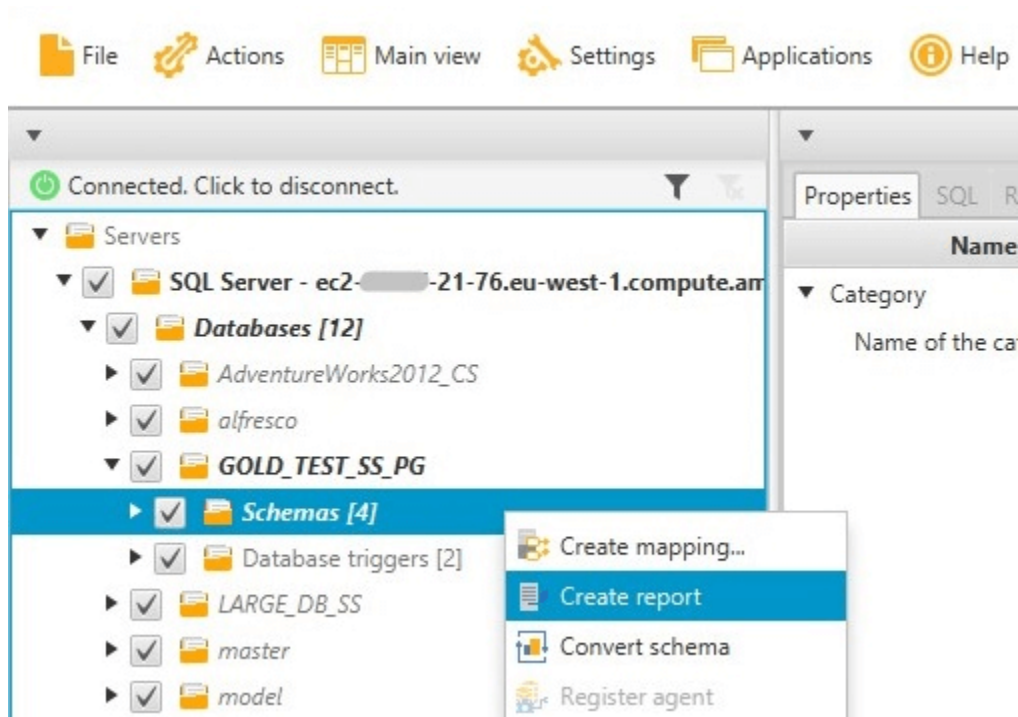
AWS Schema Conversion Tool Membuat laporan penilaian migrasi database untuk membantu Anda mengonversi skema. Laporan penilaian migrasi database memberikan informasi penting tentang konversi skema dari database sumber Anda ke database target Anda. Laporan ini merangkum semua tugas konversi skema dan merinci item tindakan untuk skema yang tidak dapat dikonversi ke mesin DB database target Anda. Laporan ini juga mencakup perkiraan jumlah upaya yang diperlukan untuk menulis kode setara dalam database target Anda yang tidak dapat dikonversi secara otomatis.

Membuat laporan penilaian migrasi database

Gunakan prosedur berikut untuk membuat laporan penilaian migrasi database.

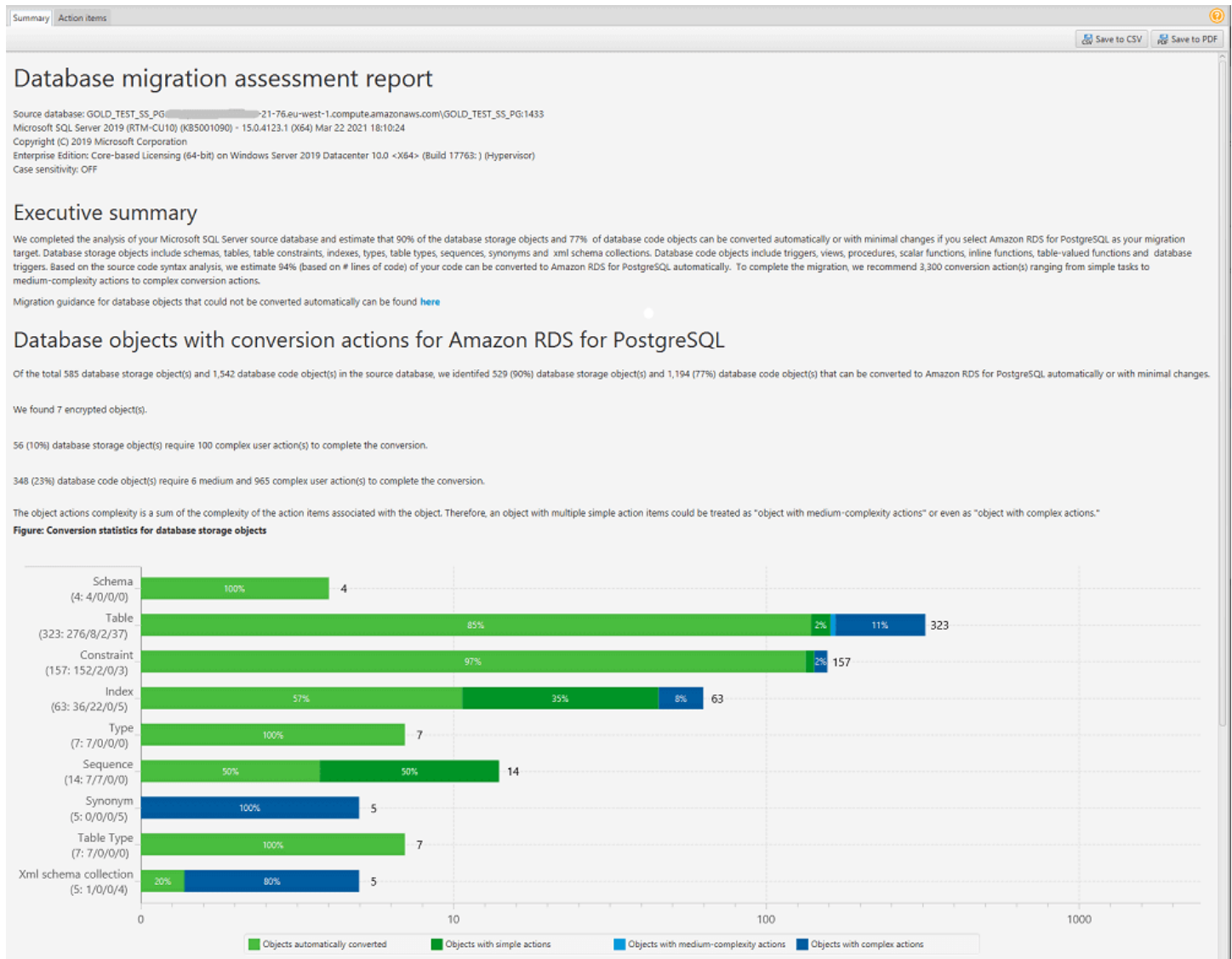
Membuat laporan penilaian migrasi database

1. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih objek skema untuk membuat laporan penilaian.
2. Buka menu konteks (klik kanan) untuk objek, lalu pilih Create Report.



Ringkasan laporan penilaian

Setelah membuat laporan penilaian, tampilan laporan penilaian akan terbuka, menampilkan tab Ringkasan. Tab Ringkasan menampilkan informasi ringkasan dari laporan penilaian migrasi database. Ini menunjukkan item yang dikonversi secara otomatis, dan item yang tidak dikonversi secara otomatis.



Untuk item skema yang tidak dapat dikonversi secara otomatis ke mesin database target, ringkasan mencakup perkiraan upaya yang diperlukan untuk membuat item skema dalam instans DB target Anda yang setara dengan yang ada di sumber Anda.

Laporan ini mengkategorikan perkiraan waktu untuk mengonversi item skema ini sebagai berikut:

- Sederhana - Tindakan yang dapat diselesaikan dalam waktu kurang dari satu jam.
- Medium - Tindakan yang lebih kompleks dan dapat diselesaikan dalam satu hingga empat jam.
- Signifikan - Tindakan yang sangat kompleks dan membutuhkan waktu lebih dari empat jam untuk menyelesaikannya.

Item tindakan laporan penilaian

Tampilan laporan penilaian juga mencakup tab Item Tindakan. Tab ini berisi daftar item yang tidak dapat dikonversi secara otomatis ke mesin database database target Anda. Jika Anda memilih item tindakan dari daftar, AWS SCT sorot item dari skema Anda yang berlaku untuk item tindakan.

Laporan ini juga berisi rekomendasi tentang cara mengonversi item skema secara manual. Untuk informasi selengkapnya tentang memutuskan cara menangani konversi manual, lihat [Menangani konversi manual di 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:** Shows a tree view of servers and databases. The selected server is 'SQL Server - ec2-21-76.eu-west-1.compute'. Underneath, there are databases like 'AdventureWorks2012_CS', 'alfresco', 'GOLD_TEST_SS_PG', 'LARGE_DB_SS', 'master', 'model', 'msdb', 'tempdb', and 'TEST'. Schemas include 'dbo' with tables like 'MSSQL_TemporalHistoryFor_1013578', 'MSSQL_TemporalHistoryFor_9655784', 'NonPartitionTable', 'PartitionTable', 'Position', 'test', 'tmp_tbl_sys_ver (System-Versioned)', and 'tmp_tbl_sys_ver_alter (System-Version)'. There are also Graph Tables, External Tables, Views, Procedures, and various SQL functions.
- 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. After the INSERT operation, you can make use of the rows saved in the temporary table. Number of occurrences: 1 | Documentation reference(s): <http://dev.mysql.com/doc/refman/8.0/en/insert.html>
 - 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. Parameter: @InputPosNo (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 | Documentation reference(s): <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
 - 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
- Source Microsoft SQL Server procedure: POSITION_UPDATE_CASH_CGT_BULK:**

```

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:**

Name	Value
Category	Schemas
Name of the category	Schemas

Menyimpan laporan penilaian

Anda dapat menyimpan salinan lokal dari laporan penilaian migrasi database sebagai file PDF atau file nilai dipisahkan koma (CSV). File CSV hanya berisi informasi item tindakan. File PDF berisi informasi ringkasan dan item tindakan, seperti yang ditunjukkan pada contoh berikut.

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.

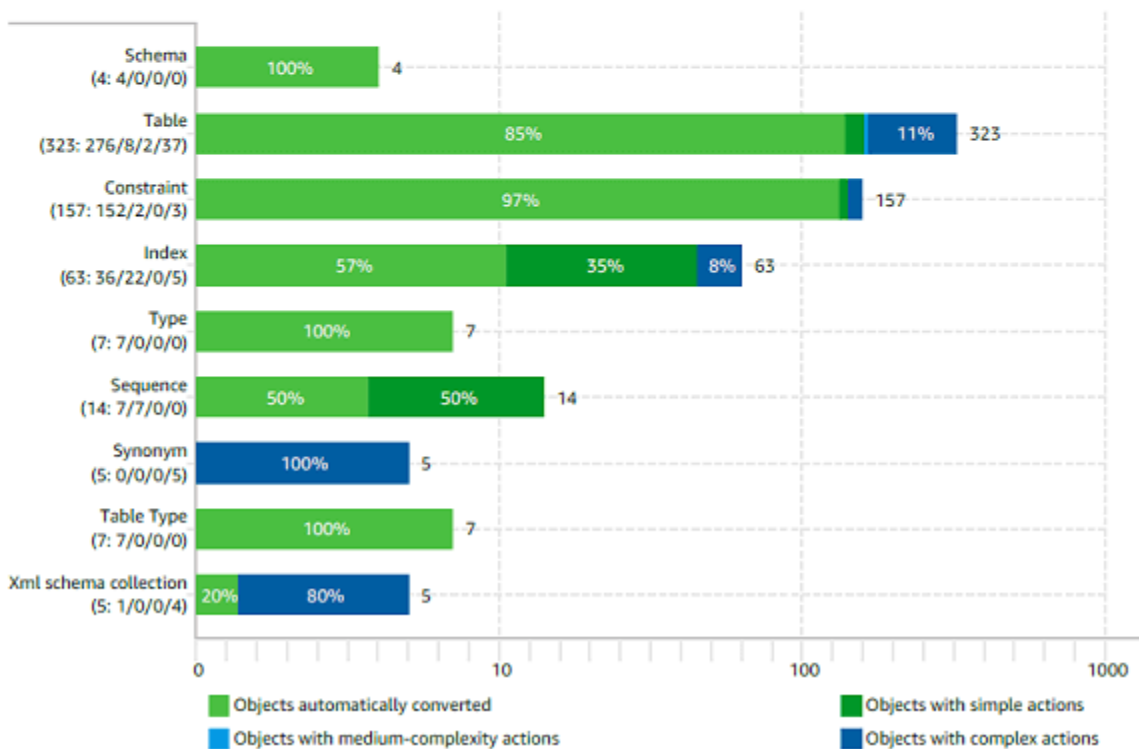
We found 7 encrypted object(s).

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



Menangani konversi manual di AWS SCT

Laporan penilaian mencakup daftar item yang tidak dapat dikonversi secara otomatis ke mesin database database target Anda. Untuk setiap item yang tidak dapat dikonversi, ada item tindakan di Item Tindakan tab.

Anda dapat menanggapi item tindakan dalam laporan penilaian dengan cara berikut:

- Memodifikasi skema database sumber Anda.

- Memodifikasi skema database target Anda.

Memodifikasi skema sumber

Untuk beberapa item, mungkin lebih mudah untuk memodifikasi skema database dalam database sumber Anda ke skema yang dapat dikonversi secara otomatis. Pertama, verifikasi bahwa perubahan baru kompatibel dengan arsitektur aplikasi Anda, lalu perbarui skema di database sumber Anda. Terakhir, segarkan proyek Anda dengan informasi skema yang diperbarui. Anda kemudian dapat mengonversi skema yang diperbarui, dan membuat laporan penilaian migrasi database baru. Item tindakan tidak lagi muncul untuk item yang berubah dalam skema sumber.

Keuntungan dari proses ini adalah skema yang diperbarui selalu tersedia saat Anda menyegarkan dari database sumber Anda.

Memodifikasi skema target

Untuk beberapa item, mungkin lebih mudah untuk menerapkan skema yang dikonversi ke database target Anda, dan kemudian menambahkan item skema yang setara secara manual ke database target Anda untuk item yang tidak dapat dikonversi secara otomatis. Anda dapat menulis semua skema yang dapat dikonversi secara otomatis ke database target Anda dengan menerapkan skema. Untuk informasi selengkapnya, lihat [Menyimpan dan menerapkan skema yang telah dikonversi AWS SCT](#).

Skema yang ditulis ke database target Anda tidak berisi item yang tidak dapat dikonversi secara otomatis. Setelah menerapkan skema ke database target Anda, Anda kemudian dapat secara manual membuat skema dalam database target Anda yang setara dengan yang ada di database sumber. Item tindakan dalam laporan penilaian migrasi database berisi saran tentang cara membuat skema yang setara.

Warning

Jika Anda membuat skema secara manual dalam database target Anda, simpan salinan pekerjaan manual apa pun yang Anda lakukan. Jika Anda menerapkan skema yang dikonversi dari proyek Anda ke database target Anda lagi, itu akan menimpa pekerjaan manual yang telah Anda lakukan.

Dalam beberapa kasus, Anda tidak dapat membuat skema setara dalam database target Anda. Anda mungkin perlu merancang ulang sebagian aplikasi dan database Anda untuk menggunakan fungsionalitas yang tersedia dari mesin untuk database target Anda. Dalam kasus lain, Anda dapat mengabaikan skema yang tidak dapat dikonversi secara otomatis.

Memperbarui dan menyegarkan skema yang telah dikonversi AWS SCT

Anda dapat memperbarui skema sumber dan skema target dalam proyek Anda AWS Schema Conversion Tool.

- **Sumber** - Jika Anda memperbarui skema untuk database sumber Anda, AWS SCT gantikan skema dalam proyek Anda dengan skema terbaru dari database sumber Anda. Dengan menggunakan fungsionalitas ini, Anda dapat memperbarui proyek Anda jika perubahan telah dilakukan pada skema database sumber Anda.
- **Target** - Jika Anda memperbarui skema untuk database target Anda, AWS SCT gantikan skema dalam proyek Anda dengan skema terbaru dari database target Anda. Jika Anda belum menerapkan skema apa pun ke database target Anda, AWS SCT hapus skema yang dikonversi dari proyek Anda. Anda kemudian dapat mengkonversi skema dari database sumber Anda untuk database target bersih.

Anda memperbarui skema dalam AWS SCT proyek Anda dengan memilih Refresh dari database.

Menyimpan dan menerapkan skema yang telah dikonversi AWS SCT

Ketika AWS Schema Conversion Tool menghasilkan skema dikonversi (seperti yang ditunjukkan dalam [Mengonversi skema Anda menggunakan AWS SCT](#)), itu tidak segera menerapkan skema dikonversi ke database target. Sebaliknya, skema dikonversi disimpan secara lokal dalam proyek Anda sampai Anda siap untuk menerapkannya ke database target. Menggunakan fungsi ini, Anda dapat bekerja dengan item skema yang tidak dapat dikonversi secara otomatis ke mesin database target Anda. Untuk informasi selengkapnya tentang item yang tidak dapat dikonversi secara otomatis, lihat [Membuat laporan penilaian migrasi dengan AWS SCT](#).

Anda opsional dapat memiliki alat menyimpan skema Anda dikonversi ke file sebagai script SQL sebelum menerapkan skema ke database target Anda. Anda juga dapat memiliki alat menerapkan skema dikonversi langsung ke database target Anda.

Menyimpan skema yang telah dikonversi ke file

Anda dapat menyimpan skema yang telah dikonversi sebagai skrip SQL dalam file teks. Dengan menggunakan pendekatan ini, Anda dapat memodifikasi skrip SQL yang dihasilkan dari AWS SCT ke item alamat yang alat tidak dapat dikonversi secara otomatis. Anda kemudian dapat menjalankan skrip yang diperbarui pada instans DB target Anda untuk menerapkan skema yang telah dikonversi ke database target Anda.

Untuk menyimpan skema Anda dikonversi sebagai skrip SQL

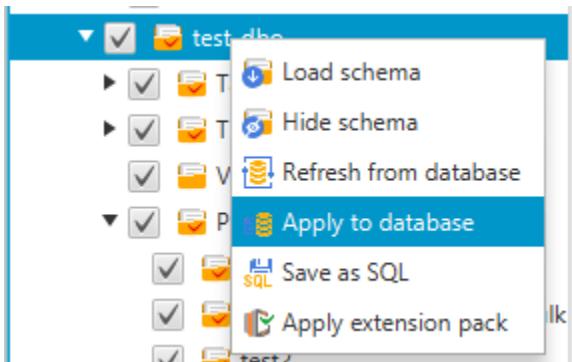
1. Pilih skema Anda dan buka menu konteks (klik kanan).
2. Pilih Simpan sebagai SQL.
3. Masukkan nama file dan pilih Simpan.
4. Simpan skema Anda yang telah dikonversi menggunakan salah satu opsi berikut:
 - Berkas tunggal
 - File tunggal per tahap
 - File tunggal per pernyataan

Untuk memilih format skrip SQL

1. Pada menu Pengaturan, pilih Pengaturan proyek.
2. Pilih Simpan skrip.
3. Untuk Vendor, pilih platform database.
4. Untuk Simpan skrip SQL, pilih bagaimana Anda ingin menyimpan skrip skema database Anda.
5. Pilih OK untuk menyimpan pengaturan.

Menerapkan skema yang telah dikonversi

Ketika Anda siap untuk menerapkan skema Anda dikonversi ke database target Anda, pilih elemen skema dari panel kanan proyek Anda. Buka konteks (klik kanan) menu untuk elemen skema, dan kemudian pilih Terapkan ke database, seperti yang ditunjukkan berikut.



Skema paket ekstensi

Pertama kali Anda menerapkan skema yang dikonversi ke instans DB target Anda, AWS SCT menambahkan skema tambahan ke instans DB target Anda. Skema ini mengimplementasikan fungsi sistem dari database sumber yang diperlukan saat menulis skema yang dikonversi ke instans DB target Anda. Skema ini disebut skema paket ekstensi.

Jangan memodifikasi skema paket ekstensi, atau Anda mungkin menemukan hasil yang tidak terduga dalam skema yang dikonversi yang ditulis ke instans DB target Anda. Ketika skema Anda sepenuhnya dimigrasi ke instans DB target Anda, dan Anda tidak perlu lagi AWS SCT, Anda dapat menghapus skema paket ekstensi.

Skema paket ekstensi diberi nama sesuai dengan database sumber Anda sebagai berikut:

- Greenplum: `aws_greenplum_ext`
- Server Microsoft SQL: `aws_sqlserver_ext`
- Netezza: `aws_netezza_ext`
- Oracle: `aws_oracle_ext`
- Kepingan salju: `aws_snowflake_ext`
- Teradata: `aws_teradata_ext`
- Vertika: `aws_vertica_ext`

Untuk informasi selengkapnya, lihat [Menggunakan paket AWS SCT ekstensi](#).

Pustaka Python

Untuk membuat fungsi kustom di Amazon Redshift, Anda menggunakan bahasa Python. Gunakan paket AWS SCT ekstensi untuk menginstal pustaka python untuk database Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Menggunakan paket AWS SCT ekstensi](#).

Mengoptimalkan Amazon Redshift dengan menggunakan AWS SCT

Anda dapat menggunakan AWS Schema Conversion Tool untuk mengoptimalkan database Amazon Redshift Anda. Menggunakan database Amazon Redshift Anda sebagai sumber, dan menguji database Amazon Redshift sebagai target, AWS SCT merekomendasikan kunci sortir dan kunci distribusi untuk mengoptimalkan database Anda.

Mengoptimalkan database Amazon Redshift Anda

Gunakan prosedur berikut untuk mengoptimalkan database Amazon Redshift Anda.

Untuk mengoptimalkan database Amazon Redshift Anda

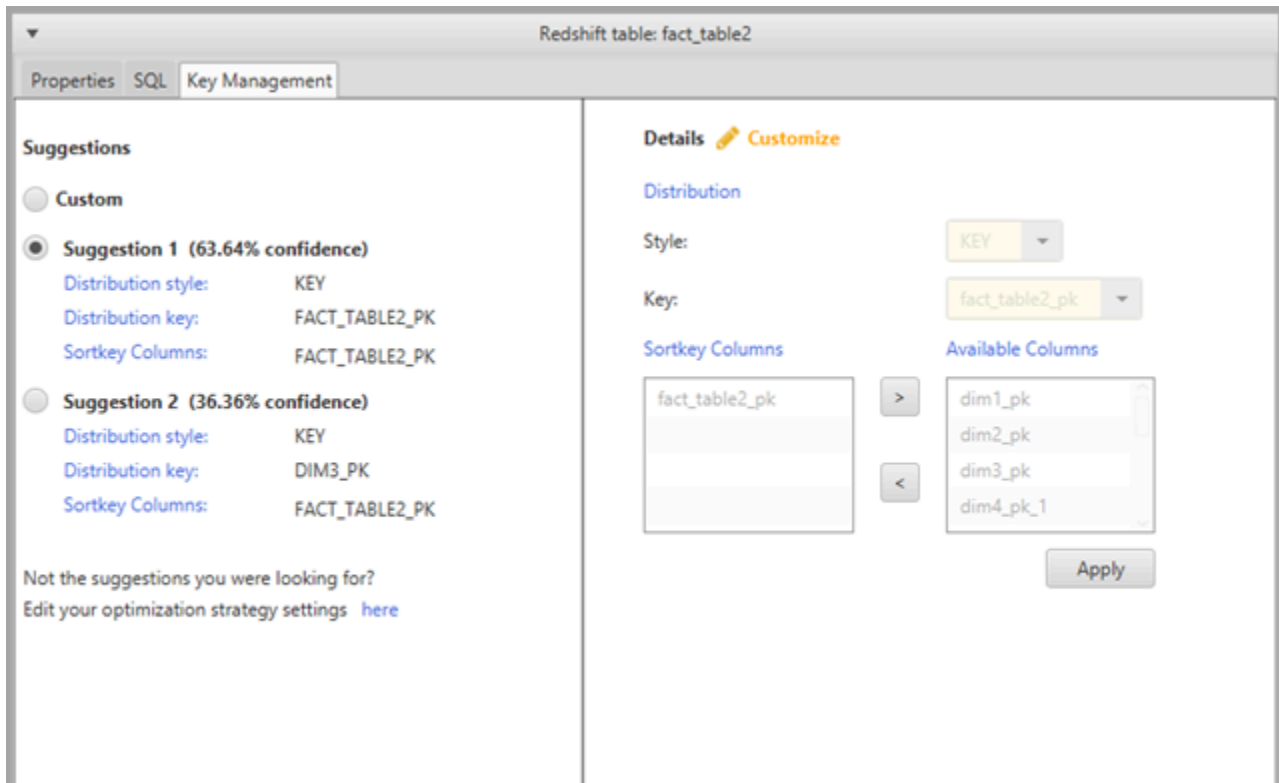
1. Ambil snapshot manual klaster Amazon Redshift Anda sebagai cadangan. Anda dapat menghapus snapshot setelah selesai mengoptimalkan klaster Amazon Redshift dan menguji setiap perubahan yang Anda buat. Untuk informasi selengkapnya, lihat [snapshot Amazon Redshift](#).
2. Pilih objek skema untuk dikonversi dari panel kiri proyek Anda. Buka menu konteks (klik kanan) untuk objek, lalu pilih Kumpulkan Statistik.

AWS SCT menggunakan statistik untuk membuat saran untuk kunci sortir dan kunci distribusi.

3. Pilih objek skema untuk dioptimalkan dari panel kiri proyek Anda. Buka menu konteks (klik kanan) untuk objek, lalu pilih Jalankan Optimasi.

AWS SCT membuat saran untuk kunci sortir dan kunci distribusi.

4. Untuk meninjau saran, perluas node tabel di bawah skema Anda di panel kiri proyek Anda, lalu pilih tabel. Pilih tab Key Management seperti yang ditunjukkan berikut ini.



Panel kiri berisi saran utama, dan menyertakan peringkat kepercayaan untuk setiap saran. Anda dapat memilih salah satu saran, atau Anda dapat menyesuaikan kunci dengan mengeditnya di panel kanan.

5. Anda dapat membuat laporan yang berisi saran pengoptimalan. Untuk membuat laporan, lakukan hal berikut:

- a. Pilih objek skema yang Anda optimalkan dari panel kiri proyek Anda. Buka menu konteks (klik kanan) untuk objek, lalu pilih Create Report.

Laporan terbuka di jendela utama, dan tab Ringkasan muncul. Jumlah objek dengan saran pengoptimalan muncul di laporan.

- b. Pilih Item Tindakan tab untuk melihat saran utama dalam format laporan.
- c. Anda dapat menyimpan salinan lokal dari laporan pengoptimalan sebagai file PDF atau file nilai dipisahkan koma (CSV). File CSV hanya berisi informasi item tindakan. File PDF berisi informasi ringkasan dan item tindakan.

6. Untuk menerapkan optimasi yang disarankan ke database Anda, pilih objek di panel kanan proyek Anda. Buka menu konteks (klik kanan) untuk objek, lalu pilih Apply to database.

Mengkonversi ekstrak, transformasi, dan beban (ETL) proses dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk bermigrasi ekstrak, transformasi, dan beban (ETL) proses. Jenis migrasi ini mencakup konversi logika bisnis terkait ETL. Logika ini dapat berada di dalam gudang data sumber Anda atau skrip eksternal yang Anda jalankan secara terpisah.

Saat ini, AWS SCT mendukung konversi skrip ETL ke objek ke AWS Glue dan Amazon Redshift RSQL, seperti yang ditunjukkan pada tabel berikut.

Sumber	Target
Skrip ETL Informatica	Informatica
Paket ETL Microsoft SQL Server Integrasi Services (SSIS)	AWS Glue atau AWS Glue Studio
Script shell dengan perintah tertanam dari Teradata Dasar Teradata Query (BTEQ)	Amazon Redshift
Skrip ETL Teradata BTEQ	AWS Glue atau Amazon Redshift RSQL
Teradata FastExport skrip pekerjaan	Amazon Redshift
Teradata FastLoad skrip pekerjaan	Amazon Redshift
Teradata MultiLoad skrip pekerjaan	Amazon Redshift

Topik

- [Mengkonversi proses ETL ke AWS Glue bersama AWS SCT](#)
- [Mengkonversi proses ETL menggunakan Python API untuk AWS Glue bersama AWS SCT](#)
- [Mengkonversi skrip Informatica ETL dengan AWS SCT](#)
- [Mengonversi SSIS ke AWS Glue bersama AWS SCT](#)
- [Mengonversi SSIS ke AWS Glue Studio bersama AWS SCT](#)

- [Mengonversi skrip Teradata BTEQ ke Amazon Redshift RSQL dengan AWS SCT](#)
- [Mengonversi skrip shell dengan perintah Teradata BTEQ yang disematkan ke Amazon Redshift RSQL dengan AWS SCT](#)
- [Konversi Teradata FastExport skrip pekerjaan untuk Amazon Redshift RSQL dengan AWS SCT](#)
- [Konversi Teradata FastLoad skrip pekerjaan untuk Amazon Redshift RSQL dengan AWS SCT](#)
- [Konversi Teradata MultiLoad skrip pekerjaan untuk Amazon Redshift RSQL dengan AWS SCT](#)

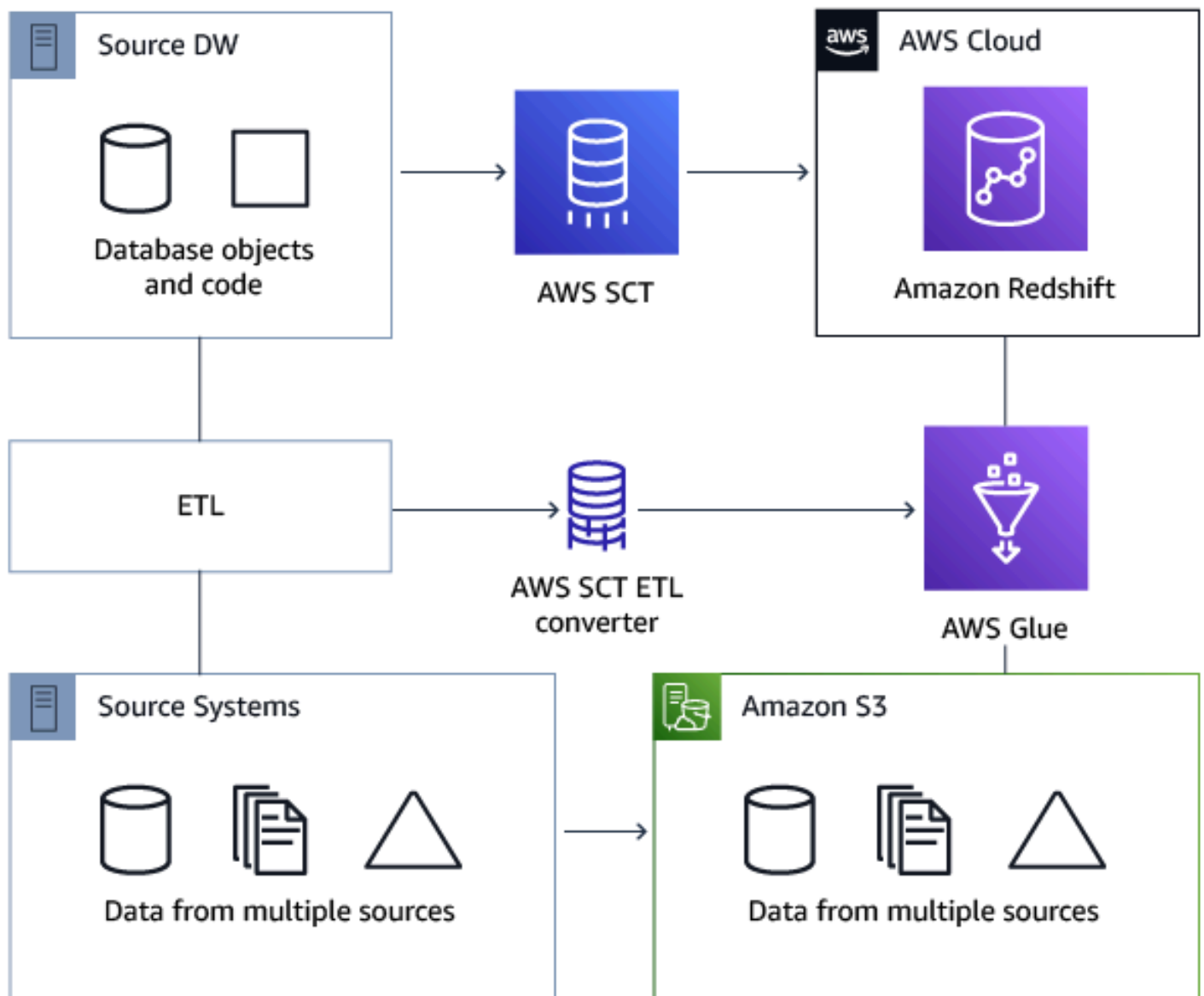
Mengkonversi proses ETL ke AWS Glue bersama AWS SCT

Berikut, Anda dapat menemukan garis besar proses untuk mengkonversi skrip ETL ke AWS Glue bersama AWS SCT. Untuk contoh ini, kami mengonversi database Oracle ke Amazon Redshift, bersama dengan proses ETL yang digunakan dengan database sumber dan gudang data.

Topik

- [Prasyarat](#)
- [Memahami AWS Glue Katalog Data](#)
- [Batasan untuk mengkonversi menggunakan AWS SCT bersama AWS Glue](#)
- [Langkah 1: Buat proyek baru](#)
- [Langkah 2: Buat AWS Glue pekerjaan](#)

Diagram arsitektur berikut menunjukkan contoh proyek migrasi database yang mencakup konversi skrip ETL ke AWS Glue.



Prasyarat

Sebelum memulai, lakukan hal berikut:

- Migrasikan database sumber apa pun yang ingin Anda migrasi ke AWS.
- Migrasikan gudang data target ke AWS.
- Kumpulkan daftar semua kode yang terlibat dalam proses ETL Anda.
- Kumpulkan daftar semua informasi koneksi yang diperlukan untuk setiap database.

Juga, AWS Glue membutuhkan izin untuk mengakses lainnya AWS sumber daya atas nama Anda. Anda memberikan izin tersebut dengan menggunakan AWS Identity and Access Management (IAM). Pastikan Anda membuat kebijakan IAM untuk AWS Glue. Untuk informasi lebih lanjut, lihat [Buat kebijakan IAM untuk AWS Glue service](#) di dalam AWS Glue Panduan Pengembang.

Memahami AWS Glue Katalog Data

Sebagai bagian dari proses konversi, AWS Glue memuat informasi mengenai sumber dan target database. Ini mengatur informasi ini ke dalam kategori, dalam struktur yang disebut apohon. Strukturnya meliputi:

- Koneksi- Parameter koneksi
- Crawler- Daftar crawler, satu crawler untuk setiap skema
- Database- Wadah yang menampung tabel
- Tabel- Definisi metadata yang mewakili data dalam tabel
- Lowongan kerja ETL- Logika bisnis yang melakukan pekerjaan ETL
- Pemicu- Logika yang mengontrol ketika pekerjaan ETL berjalan di AWS Glue (apakah sesuai permintaan, berdasarkan jadwal, atau dipicu oleh acara pekerjaan)

Yang AWS Glue Katalog Data adalah indeks ke lokasi, skema, dan metrik runtime data Anda. Saat Anda bekerja dengan AWS Glue dan AWS SCT, yang AWS Glue Katalog Data berisi referensi ke data yang digunakan sebagai sumber dan target pekerjaan ETL Anda di AWS Glue. Untuk membuat gudang data Anda, katalog data ini.

Anda menggunakan informasi dalam Katalog Data untuk membuat dan memantau tugas ETL Anda. Biasanya, Anda menjalankan crawler untuk mengambil inventaris data di penyimpanan data Anda, tetapi ada cara lain untuk menambahkan tabel metadata ke dalam Katalog Data Anda.

Ketika Anda menentukan tabel di Katalog Data Anda, Anda menambahkannya ke database. Sebuah database digunakan untuk mengatur tabel di AWS Glue.

Batasan untuk mengkonversi menggunakan AWS SCT bersama AWS Glue

Batasan berikut berlaku saat mengonversi menggunakan AWS SCT bersama AWS Glue.

Sumber Daya	Batas default
-------------	---------------

Jumlah database untuk setiap akun	10.000
Jumlah tabel untuk setiap database	100.000
Jumlah partisi untuk setiap tabel	1.000.000
Jumlah versi tabel untuk setiap tabel	100.000
Jumlah tabel untuk setiap akun	1.000.000
Jumlah partisi untuk setiap akun	10.000.000
Jumlah versi tabel untuk setiap akun	1.000.000
Jumlah koneksi untuk setiap akun	1.000
Jumlah crawler untuk setiap akun	25
Jumlah pekerjaan untuk setiap akun	25
Jumlah pemicu untuk setiap akun	25
Jumlah pekerjaan bersamaan yang dijalankan untuk setiap akun	30
Jumlah pekerjaan bersamaan berjalan untuk setiap pekerjaan	3
Jumlah pekerjaan untuk setiap pemicu	10
Jumlah endpoint pengembangan untuk setiap akun	5
Unit pemrosesan data maksimum (DPU) yang digunakan oleh titik akhir pengembangan pada satu waktu	5
DPU maksimum yang digunakan oleh peran pada satu waktu	100

Panjang nama basis data	<p>Tidak terbatas.</p> <p>Untuk kompatibilitas dengan toko metadata lainnya, seperti Apache Hive, nama diubah untuk menggunakan karakter huruf kecil.</p> <p>Jika Anda berencana untuk mengakses database dari Amazon Athena, berikan nama hanya dengan karakter alfanumerik dan garis bawah.</p>
Panjang nama koneksi	Tidak terbatas.
Panjang nama crawler	Tidak terbatas.

Langkah 1: Buat proyek baru

Untuk membuat proyek baru, ambil langkah-langkah tingkat tinggi ini:

1. Membuat proyek baru di AWS SCT. Untuk informasi selengkapnya, lihat [Membuat AWS SCT proyek](#).
2. Tambahkan sumber dan basis data target Anda ke proyek. Untuk informasi selengkapnya, lihat [Menambahkan server database ke AWS SCT proyek](#).

Pastikan bahwa Anda telah memilih Gunakan AWS Glue dalam pengaturan koneksi database target. Untuk melakukannya, pilih AWS Glue tab. Untuk Salin dari AWS profil, pilih profil yang ingin Anda gunakan. Profil harus secara otomatis mengisi AWS kunci akses, kunci rahasia, dan folder bucket Amazon S3. Jika tidak, masukkan informasi ini sendiri. Setelah Anda memilih OKE, AWS Glue menganalisis objek dan beban metadata ke AWS Glue Katalog Data.

Bergantung pada pengaturan keamanan, Anda mungkin mendapatkan pesan peringatan yang mengatakan bahwa akun Anda tidak memiliki hak istimewa yang memadai untuk beberapa skema di server. Jika Anda memiliki akses ke skema yang Anda gunakan, Anda dapat mengabaikan pesan ini dengan aman.

3. Untuk menyelesaikan persiapan untuk mengimpor ETL Anda, sambungkan ke sumber dan basis data target Anda. Untuk melakukannya, pilih database Anda di pohon metadata sumber atau target, lalu pilih Hubungkan ke server.

AWS Glue menciptakan database pada server database sumber dan satu di server database target untuk membantu dengan konversi ETL. Database pada server target berisi AWS Glue Katalog Data. Untuk menemukan objek tertentu, gunakan pencarian di panel sumber atau target.

Untuk melihat bagaimana objek tertentu mengkonversi, menemukan item yang ingin Anda konversi, dan pilih Mengkonversi skemadari menu konteksnya (klik kanan). AWS SCT mengubah objek yang dipilih ini menjadi script.

Anda dapat meninjau skrip yang dikonversi dari Skrip folder di panel kanan. Saat ini, script adalah objek virtual, yang tersedia hanya sebagai bagian dari Anda AWS SCT proyek.

Untuk membuat AWS Glue pekerjaan dengan skrip Anda yang telah dikonversi, unggah skrip Anda ke Amazon S3. Untuk mengunggah skrip ke Amazon S3, pilih skrip, lalu pilih Simpan ke S3 dari menu konteksnya (klik kanan).

Langkah 2: Buat AWS Glue pekerjaan

Setelah Anda menyimpan skrip ke Amazon S3, Anda dapat memilihnya dan kemudian memilih Mengkonfigurasi AWS Glue Pekerjaan untuk membuka wizard untuk mengkonfigurasi AWS Glue pekerjaan. Wizard membuatnya lebih mudah untuk mengatur ini:

1. Pada tab pertama wizard, Aliran Data Desain, Anda dapat memilih strategi eksekusi dan daftar skrip yang ingin Anda sertakan dalam pekerjaan yang satu ini. Anda dapat memilih parameter untuk setiap skrip. Anda juga dapat mengatur ulang skrip sehingga mereka berjalan dalam urutan yang benar.
2. Pada tab kedua, Anda dapat memberi nama pekerjaan Anda, dan langsung mengkonfigurasi pengaturan untuk AWS Glue. Pada layar ini, Anda dapat mengkonfigurasi pengaturan berikut:
 - AWS Identity and Access Management (IAM) peran
 - Nama file skrip dan jalur file
 - Mengenkripsi skrip menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3)
 - Direktori sementara
 - Jalur pustaka Python yang dihasilkan
 - Jalur pustaka Python pengguna
 - Path untuk file.jar dependen
 - Path file yang direferensikan
 - DPU serentak untuk setiap pekerjaan yang dijalankan

- Konkurensi maksimum
- Batas waktu kerja (dalam menit)
- Ambang batas pemberitahuan penundaan (dalam hitungan menit)
- Jumlah percobaan
- Konfigurasi keamanan
- Enkripsi sisi server

3. Pada langkah ketiga, atau tab, Anda memilih koneksi yang dikonfigurasi ke titik akhir target.

Setelah Anda selesai mengkonfigurasi pekerjaan, itu akan ditampilkan di bawah pekerjaan ETL diAWS GlueKatalog Data. Jika Anda memilih pekerjaan, pengaturan akan ditampilkan sehingga Anda dapat meninjau atau mengeditnya. Untuk membuat pekerjaan baru diAWS Glue, pilihBuatAWS GluePekerjaandari konteks (klik kanan) menu untuk pekerjaan. Melakukan hal ini berlaku definisi skema. Untuk menyegarkan tampilan, pilihRefresh dari databasedari menu konteks (klik kanan).

Pada titik ini, Anda dapat melihat pekerjaan Anda diAWS Gluekonsol. Untuk melakukannya, masuk keAWS Management Console dan buka/bukalahAWS Gluekonsol di<https://console.aws.amazon.com/glue/>.

Anda dapat menguji pekerjaan baru untuk memastikan bahwa itu bekerja dengan benar. Untuk melakukannya, periksa dulu data di tabel sumber Anda, lalu verifikasi bahwa tabel target kosong. Jalankan pekerjaan, dan periksa lagi. Anda dapat melihat log kesalahan dariAWS Gluekonsol.

Mengkonversi proses ETL menggunakan Python API untukAWS GluebersamaAWS SCT

Di bagian berikut, Anda dapat menemukan deskripsi konversi yang memanggilAWS GlueOperasi API di Python. Untuk informasi lebih lanjut, lihat[ProgramAWS GlueScript ETL dalam Python](#)di dalam AWS GluePanduan Pengembang.

Topik

- [Langkah 1: Buat database](#)
- [Langkah 2: Buat koneksi](#)
- [Langkah 3: BuatAWS Glueperayap](#)

Langkah 1: Buat database

Langkah pertama adalah untuk membuat database baru dalam AWS Glue Katalog Data dengan menggunakan [AWSAPI SDK](#). Ketika Anda menentukan tabel di Katalog Data, Anda menambahkannya ke database. Sebuah database digunakan untuk mengatur tabel di AWS Glue.

Contoh berikut menunjukkan `create_database` metode Python API untuk AWS Glue.

```
response = client.create_database(  
    DatabaseInput={  
        'Name': 'database_name',  
        'Description': 'description',  
        'LocationUri': 'string',  
        'Parameters': {  
            'parameter-name': 'parameter value'  
        }  
    }  
)
```

Jika Anda menggunakan Amazon Redshift, nama database dibentuk sebagai berikut.

```
{redshift_cluster_name}_{redshift_database_name}_{redshift_schema_name}
```

Nama lengkap kluster Amazon Redshift untuk contoh ini adalah sebagai berikut.

```
rsdbb03.apq1mpqso.us-west-2.redshift.amazonaws.com
```

Berikut ini menunjukkan contoh nama database yang terbentuk dengan baik. Dalam hal ini `rsdbb03` adalah nama, yang merupakan bagian pertama dari nama lengkap dari endpoint cluster. Database diberi nama `dev` dan skemanya adalah `ora_glue`.

```
rsdbb03_dev_ora_glue
```

Langkah 2: Buat koneksi

Buat koneksi baru dalam Katalog Data dengan menggunakan [AWSAPI SDK](#).

Contoh berikut menunjukkan menggunakan [create_connection](#) metode Python API untuk 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'  
            ]  
        }  
    }  
)
```

Parameter yang digunakan dalam `create_connection` adalah sebagai berikut:

- **Name**(UTF-8 string) - diperlukan. Untuk Amazon Redshift, nama koneksi dibentuk sebagai berikut: `Redshift_<Endpoint-name>_<redshift-database-name>`, misalnya: `Redshift_abcde03_dev`
- **Description**(UTF-8 string) - Deskripsi koneksi Anda.
- **ConnectionType**(UTF-8 string) - Diperlukan. Jenis koneksi. Saat ini, hanya JDBC yang didukung; SFTP tidak didukung.
- **ConnectionProperties**(dict) - Diperlukan. Daftar pasangan kunci-nilai yang digunakan sebagai parameter untuk koneksi ini, termasuk URL koneksi JDBC, nama pengguna, dan kata sandi.
- **PhysicalConnectionRequirements**(dict) - Persyaratan koneksi fisik, yang meliputi:
 - **SubnetId**(UTF-8 string) - ID subnet yang digunakan oleh koneksi.
 - **SecurityGroupIdList**(daftar) - Daftar ID grup keamanan yang digunakan oleh koneksi.
 - **AvailabilityZone**(UTF-8 string) - Diperlukan. Availability Zone yang berisi endpoint. Parameter ini tidak lagi digunakan.

Langkah 3: BuatAWS Glueperayap

Selanjutnya, Anda membuatAWS Gluecrawler untuk mengisiAWS Gluekatalog. Untuk informasi lebih lanjut, lihat[Katalog tabel dengan crawler](#)di dalamAWS GluePanduan Pengembang.

Langkah pertama dalam menambahkan crawler adalah membuat database baru dalam Katalog Data dengan menggunakan[AWSAPI SDK](#). Sebelum Anda mulai, pastikan untuk terlebih dahulu menghapus versi sebelumnya dengan menggunakan`delete_crawler`operasi.

Saat Anda membuat crawler, beberapa pertimbangan berlaku:

- Untuk nama crawler, gunakan format`<redshift_node_name>_<redshift_database_name>_<redshift_shema_name>`, misalnya:abcde03_dev_ora_glue
- Gunakan peran IAM yang sudah ada. Untuk informasi selengkapnya tentang cara membuat peran IAM, lihat[Membuat peran IAM](#)di dalamPanduan Pengguna IAM.
- Gunakan nama database yang Anda buat di langkah sebelumnya.
- Gunakan`ConnectionName`parameter, yang diperlukan.
- Untuk`path`parameter, menggunakan path ke target JDBC, misalnya:dev/ora_glue/%

Contoh berikut menghapus crawler yang ada dan kemudian membuat yang baru dengan menggunakan Python API untukAWS Glue.

```
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'
)

```

Buat dan kemudian jalankan crawler yang terhubung ke satu atau lebih penyimpanan data, menentukan struktur data, dan menulis tabel ke dalam Katalog Data. Anda dapat menjalankan crawler sesuai jadwal, seperti yang ditunjukkan berikut.

```

response = client.start_crawler(
    Name='string'
)

```

Contoh ini menggunakan Amazon Redshift sebagai target. Jenis data Amazon Redshift dipetakan ke AWS Glue tipe data dengan cara berikut setelah crawler berjalan.

Tipe data Amazon Redshift	AWS Glue tipe data
smallint	smallint
bilangan bulat	int
bigint	bigint

decimal	desimal (18,0)
desimal (p, s)	desimal (p, s)
nyata	double
presisi ganda	double
boolean	boolean
char	string
varchar	string
varchar (n)	string
tanggal	tanggal
timestamp	timestamp
stemamptz	timestamp

Mengkonversi skrip Informatica ETL dengan AWS SCT

Anda dapat menggunakan AWS SCT command line interface (CLI) untuk mengkonversi script Informatica ETL Anda sehingga Anda dapat menggunakan script dengan database target baru Anda. Konversi ini mencakup tiga langkah kunci. Pertama, AWS SCT mengkonversi kode SQL yang tertanam dalam objek Informatica Anda. Berikutnya, AWS SCT mengubah nama objek database sesuai dengan aturan migrasi yang Anda tentukan dalam proyek Anda. Akhirnya, AWS SCT mengalihkan koneksi skrip ETL Informatica Anda ke database target baru.

Anda dapat mengkonversi skrip ETL Informatica sebagai bagian dari AWS SCT proyek konversi database. Pastikan bahwa Anda menambahkan sumber dan target database untuk proyek ketika Anda mengkonversi script Informatica ETL.

Untuk mengonversi skrip ETL Informatica, pastikan Anda menggunakan AWS SCT versi 1.0.667 atau lebih tinggi. Juga, biasakan diri Anda dengan antarmuka baris perintah AWS SCT. Untuk informasi selengkapnya, lihat [AWS SCT Referensi CLI](#).

Untuk mengkonversi skrip ETL Informatica menggunakan AWS SCT

1. Buat yang baru AWS SCT Script CLI atau mengedit template skenario yang ada. Misalnya, Anda dapat mengunduh dan mengedit `InformaticaConversionTemplate.scts` template. Untuk informasi selengkapnya, lihat [Mendapatkan skenario CLI](#).
2. Unduh driver JDBC yang diperlukan untuk database sumber dan target Anda. Tentukan lokasi driver ini menggunakan `SetGlobalSettings` perintah. Juga, tentukan folder di mana AWS SCT dapat menyimpan file log.

Contoh kode berikut menunjukkan cara menambahkan path ke driver Oracle dan PostgreSQL ke AWS SCT pengaturan. Setelah Anda menjalankan contoh kode ini, AWS SCT menyimpan file log di `C:\sct_log` folder. Juga, AWS SCT menyimpan file log konsol di `C:\Temp\oracle_postgresql` folder.

```
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"}'
```

3. Buat yang baru AWS SCT proyek. Masukkan nama dan lokasi proyek Anda.

Contoh kode berikut menciptakan `oracle_postgresql` proyek di `C:\Temp` folder.

```
CreateProject
  -name: 'oracle_postgresql'
  -directory: 'C:\Temp'
/
```

4. Tambahkan informasi koneksi tentang sumber dan basis data target Anda.

Contoh kode berikut menambahkan database Oracle dan PostgreSQL sebagai sumber dan target untuk Anda AWS SCT proyek.

```

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'
/

```

Pada contoh sebelumnya, ganti *source_user* dan *target_pengguna* dengan nama-nama pengguna database Anda. Selanjutnya, ganti *source_kata_sandi* dan *target_kata_sandi* dengan kata sandi Anda. Untuk *source_address* dan *target_alamat*, masukkan alamat IP server database sumber dan target Anda.

Untuk terhubung ke database Oracle versi 19 dan yang lebih tinggi, gunakan nama layanan Oracle di `AddSource` perintah. Untuk melakukannya, tambahkan `-connectionType` parameter dan mengatur nilainya ke `'basic_service_name'`. Kemudian, tambahkan `-serviceName` parameter dan mengatur nilainya ke nama layanan Oracle Anda. Untuk informasi lebih lanjut tentang `AddSource` perintah, lihat [AWS Schema Conversion Tool Referensi CLI](#).

5. Buat yang baru AWS SCT aturan pemetaan, yang mendefinisikan mesin database target untuk setiap skema database sumber. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#).

Contoh kode berikut membuat aturan pemetaan yang mencakup semua skema database sumber Oracle dan mendefinisikan PostgreSQL sebagai target migrasi.

```

AddServerMapping
  -sourceTreePath: 'Servers.ORACLE'
  -targetTreePath: 'Servers.POSTGRESQL'

```


/

6. Tambahkan informasi koneksi tentang sumber Informatica dan target file XML-mu.

Contoh kode berikut menambahkan file XML-Informatica dari `C:\Informatica_source` dan `C:\Informatica_target`.

```
AddSource
  -name: 'INFA_SOURCE'
  -vendor: 'INFORMATICA'
  -mappingsFolder: 'C:\Informatica_source'
/
AddTarget
  -name: 'INFA_TARGET'
  -vendor: 'INFORMATICA'
  -mappingsFolder: 'C:\Informatica_target'
/
```

7. Buat aturan pemetaan lain untuk menentukan target file XML Informatica untuk sumber file XMLmu Informatica.

Contoh kode berikut membuat aturan pemetaan yang mencakup sumber dan target file XML-Informatica digunakan dalam contoh sebelumnya.

```
AddServerMapping
  -sourceTreePath: 'ETL.INFA_SOURCE'
  -targetTreePath: 'ETL.INFA_TARGET'
/
```

8. Tentukan koneksi server database yang sesuai dengan referensi nama koneksi Informatica.

Contoh kode berikut mengkonfigurasi pengalihan skrip ETL Informatica Anda dari sumber Anda ke database target baru. Contoh ini juga mengkonfigurasi variabel koneksi.

```
ConfigureInformaticaConnectionsRedirect
  -treePath: 'ETL.INFA_SOURCE.Files'
  -connections: '{
    "ConnectionNames": [
      {
        "name": "Oracle_src",
        "newName": "postgres",
        "treePath": "Servers.ORACLE"
```

```

}
]
"ConnectionVariables": [
{
    "name": "$Source",
    "treePath": "Servers.ORACLE"
}
]
}'
/

```

9. Konversi skema database sumber Anda dan skrip ETL Informatica.

Contoh kode berikut mengkonversi semua sumber skema database Oracle Anda dan file XML-Informatica Anda.

```

Convert
  -treePath: 'Servers.ORACLE.Schemas.%'
/
Convert
  -treePath: 'ETL.INFA_SOURCE.Files'
/

```

10. (Opsional) Simpan proyek konversi Anda dan laporan penilaian. Laporan ini mencakup item tindakan konversi dan rekomendasi tentang cara mengatasinya.

Contoh kode berikut menyimpan proyek Anda dan menyimpan salinan laporan penilaian sebagai file PDF di C:\Tempfolder.

```

SaveProject
/
SaveReportPDF
  -treePath: 'ETL.INFA_SOURCE.Files'
  -file: 'C:\Temp\Informatica.pdf'
/

```

11. Simpan file XML-Informatica Anda yang telah dikonversi.

Contoh kode berikut menyimpan file XML-dikonversi di C:\Tempfolder. Anda menentukan folder ini pada langkah sebelumnya menggunakan `AddTarget` perintah.

```

SaveTargetInformaticaXML

```

```
-treePath: 'ETL.INFA_TARGET.Files'  
/
```

12. Simpan skrip Anda sebagai `.sct` file dan menjalankannya menggunakan `RunSCTBatch` perintah di `AWS SCT CLI`. Untuk informasi selengkapnya, lihat [AWS SCT Modus skrip CLI](#).

Contoh berikut menjalankan `Informatica.sct` script di `C:\Temp` folder. Anda dapat menggunakan contoh ini di Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\Temp\Informatica.scts"
```

Jika Anda mengedit skrip ETL Informatica sumber Anda, kemudian jalankan `AWS SCT Script CLI` lagi.

Mengonversi SSIS ke AWS Glue bersama AWS SCT

Berikut ini, Anda dapat menemukan cara mengonversi paket Microsoft SQL Server Integration Services (SSIS) ke AWS Glue memakai AWS SCT.

Untuk mengonversi paket Microsoft SSIS ke AWS Glue, pastikan bahwa Anda menggunakan AWS SCT versi 1.0.642 atau lebih tinggi. Anda juga perlu memiliki proyek SSIS dengan paket ETL `-.dtsx`, `.conmgr`, dan `.paramsfile` di folder lokal.

Anda tidak memerlukan server SSIS yang diinstal. Proses konversi berjalan melalui file SSIS lokal.

Untuk mengonversi paket SSIS ke AWS Glue memakai AWS SCT

1. Buat proyek baru di `AWS SCT` atau membuka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called "Membuat proyek"](#).
2. Pilih **Tambahkan sumber** dari menu untuk menambahkan paket SSIS sumber baru ke proyek Anda.
3. Pilih **Layanan Integrasi SQL Server** dan lengkapi yang berikut ini:
 - Nama koneksi- Masukkan nama untuk koneksi Anda. `AWS SCT` menampilkan nama ini di pohon metadata.
 - Folder paket SSIS- Pilih jalur ke folder proyek SSIS Anda dengan paket.

- AWS SCT membaca file proyek (file dengan ekstensi `.dtsx`, `.conmgr` atau `.params`) dari folder lokal dan mem-parsing mereka. Kemudian mengatur mereka menjadi AWS SCT pohon kategori.
4. Pilih Tambahkan target dari menu untuk menambahkan platform target baru untuk mengonversi paket SSIS sumber Anda.
 5. Pilih AWS Glue dan lengkapi yang berikut ini:
 - Nama koneksi- Masukkan nama untuk koneksi Anda. AWS SCT menampilkan nama ini di pohon metadata.
 - Salin dari AWS profil- Pilih profil yang akan digunakan.
 - AWS kunci akses— Masukkan AWS kunci akses.
 - AWS kunci rahasia— Masukkan AWS kunci rahasia.
 - Region— Pilih Wilayah AWS yang ingin Anda gunakan dari daftar.
 - Folder bucket Amazon S3— Masukkan jalur folder untuk bucket Amazon S3 yang akan Anda gunakan.

Anda dapat menggunakan virtual AWS Glue Target. Dalam kasus ini, Anda tidak perlu menentukan kredensi koneksi. Untuk informasi selengkapnya, lihat [the section called “Target virtual”](#).

6. Buat aturan pemetaan baru yang menyertakan paket SSIS sumber Anda dan AWS Glue Target. Untuk informasi selengkapnya, lihat [the section called “Aturan baru”](#).
7. Pada Lihat menu, pilih Tampilan utama.
8. Dalam tampilan pohon SSIS, buka menu konteks (klik kanan) untuk Manajer koneksi, dan kemudian pilih Konfigurasi koneksi.
9. Konfigurasi manajer koneksi proyek.

Untuk mengkonfigurasi pemetaan koneksi untuk manajer koneksi SSIS, tentukan AWS Glue koneksi untuk manajer koneksi SSIS yang sesuai. Pastikan bahwa Anda AWS Glue koneksi sudah dibuat.

- a. Di bawah Koneksi, pilih Koneksi proyek.
 - b. Untuk Koneksi katalog lem, pilih yang sesuai AWS Glue koneksi.
10. Konfigurasi pengelola koneksi paket:
 - a. Di bawah Koneksi, pilih paket Anda.

- b. Untuk koneksi katalog lem, pilih yang sesuai AWS Glue koneksi.
 - c. Ulangi tindakan ini untuk semua koneksi yang tersedia untuk paket Anda.
11. Pilih Apply (Terapkan).
12. Konversikan paket Anda. Dalam tampilan pohon sumber, temukan Paket. Buka menu konteks (klik kanan) untuk paket Anda, lalu pilih Mengkonversi paket.
13. Simpan skrip yang telah dikonversi ke Amazon S3. Dalam tampilan pohon target, temukan Skrip paket. Buka menu konteks (klik kanan) untuk skrip yang telah dikonversi, lalu pilih Simpan ke S3.
14. Konfigurasi AWS Glue pekerjaan. Dalam tampilan pohon target, temukan Skrip paket. Buka menu konteks (klik kanan) untuk skrip yang telah dikonversi, lalu pilih Mengkonfigurasi AWS Glue pekerjaan.
15. Lengkapi tiga bagian konfigurasi:
 - a. Lengkapi Desain aliran data bagian:
 - Strategi eksekusi- Pilih bagaimana pekerjaan Anda akan menjalankan skrip ETL. Pilih SEKUENSIAL untuk menjalankan script dalam urutan yang ditentukan dalam wizard. Pilih PARALEL untuk menjalankan skrip secara paralel, mengabaikan urutan yang ditentukan dalam wizard.
 - Skrip- Pilih nama skrip Anda yang telah dikonversi.
 - Pilih Selanjutnya.
 - b. Lengkapi Properti pekerjaan bagian:
 - Nama— Masukkan nama AWS Glue pekerjaan.
 - Peran IAM- Pilih peran IAM yang digunakan untuk otorisasi ke sumber daya yang digunakan untuk menjalankan pekerjaan dan mengakses penyimpanan data.
 - Nama file skrip- Masukkan nama skrip Anda yang telah dikonversi.
 - Skrip file jalur S3- Masukkan jalur Amazon S3 ke skrip Anda yang telah dikonversi.
 - Enkripsi skrip menggunakan SSE-S3— Pilih opsi ini untuk melindungi data menggunakan enkripsi sisi server dengan kunci enkripsi terkelola Amazon S3 (SSE-S3).
 - Direktori sementara— Masukkan jalur Amazon S3 ke direktori sementara untuk hasil menengah. AWS Glue dan AWS Glue transformasi bawaan menggunakan direktori ini untuk membaca atau menulis ke Amazon Redshift.
 - AWS SCT secara otomatis menghasilkan jalur untuk pustaka Python. Anda dapat meninjau jalur ini di Jalur pustaka python yang dihasilkan. Anda tidak dapat mengedit

jalur yang dibuat secara otomatis ini. Untuk menggunakan pustaka Python tambahan, masukkan path diPath pustaka python pengguna.

- Path pustaka python pengguna- Masukkan jalur untuk pustaka Python pengguna tambahan. Pisahkan jalur Amazon S3 dengan koma.
- Jalur guci tergantung- Masukkan jalur untuk file jar tergantung. Pisahkan jalur Amazon S3 dengan koma.
- Jalur file yang direferensikan- Masukkan jalur untuk file tambahan, seperti file konfigurasi, yang diperlukan oleh skrip Anda. Pisahkan jalur Amazon S3 dengan koma.
- Kapasitas maksimum— Masukkan jumlah maksimumAWS Glueunit pemrosesan data (DPU) yang dapat dialokasikan saat pekerjaan ini berjalan. Anda dapat memasukkan bilangan bulat dari 2 hingga 100. Defaultnya adalah 2.
- Konkurensi maks- Masukkan jumlah maksimum berjalan bersamaan yang diizinkan untuk pekerjaan ini. Defaultnya adalah 1.AWS Gluemengembalikan kesalahan ketika ambang ini tercapai.
- Batas waktu kerja (menit)- Masukkan nilai batas waktu pada pekerjaan ETL Anda sebagai perlindungan terhadap pekerjaan pelarian. Default-nya adalah 2880 menit (48 jam) untuk tugas batch. Jika pekerjaan melebihi batas ini, status pekerjaan yang dijalankan berubah menjadiTIMEOUT.
- Ambang batas pemberitahuan tunda (menit)— Masukkan ambang batas dalam beberapa menit sebelumnyaAWS SCTmengirimkan pemberitahuan penundaan.
- Jumlah percobaan ulang— Masukkan jumlah kali (0-10) yangAWS Glueharus secara otomatis me-restart pekerjaan jika gagal. Pekerjaan yang mencapai batas batas waktu tidak dimulai ulang. Default-nya adalah 0.
- Pilih Selanjutnya.

c. Konfigurasi koneksi yang diperlukan:

- i. DariSemua koneksi, pilih yang dibutuhkanAWS Gluekoneksi dan menambahkannya ke daftarKoneksi yang dipilih.
- ii. Pilih Selesai.

16. Buat yang dikonfigurasiAWS Gluepekerjaan. Dalam tampilan pohon target, temukan dan luaskanLowongan ETL. Buka menu konteks (klik kanan) untuk pekerjaan ETL yang Anda konfigurasi, lalu pilihBuatAWS GluePekerjaan.

17. JalankanAWS Gluepekerjaan:

- a. Buka konsol AWS Glue di <https://console.aws.amazon.com/glue/>.

- b. Di panel navigasi, pilih Tugas.
- c. Pilih Tambah pekerjaan, dan kemudian pilih pekerjaan yang ingin Anda jalankan.
- d. Pada Aksi tab, pilih Jalankan pekerjaan.

Komponen SSIS itu AWS SCT dapat dikonversi ke AWS Glue

Anda dapat menggunakan AWS SCT untuk mengkonversi aliran data dan komponen aliran kontrol, serta kontainer, parameter, dan variabel.

Komponen aliran data yang didukung meliputi:

- Tujuan ADO NET
- Sumber ADO NET
- Agregat
- Transformasi Cache
- Transformasi Peta Karakter
- Transformasi Split Bersyarat
- Copy Kolom Transformasi
- Transformasi Konversi Data
- Transformasi Kolom Berasal
- Tujuan Excel
- Sumber Excel
- Ekspor Kolom Transformasi
- Tujuan File Datar
- Sumber File Datar
- Transformasi Pencarian Fuzzy
- Transformasi Kolom Impor
- Transformasi Pencarian
- Gabung Bergabung dengan Transformasi
- Gabungkan Transformasi
- Transformasi Multicast
- Tujuan ODBC

- Sumber ODBC
- Transformasi Perintah OLE DB
- Tujuan OLE DB
- Sumber OLE DB
- Transformasi Persentase Sampling
- Transformasi Pivot
- Tujuan File Mentah
- Sumber File Mentah
- RecordSetTujuan
- Transformasi Hitungan Baris
- Transformasi Pengambilan Sampel Baris
- Urutkan Transformasi
- Tujuan SQL Server
- Persatuan Semua Transformasi
- Transformasi Unpivot
- XML-Sumber

Komponen aliran kontrol yang didukung meliputi:

- Bulk Insert Task
- Execute Package Task
- Execute SQL Task
- Execute T-SQL Statement Task
- Ekspresi Tugas
- Tugas Sistem File
- Notify Operator Task
- Kirim Tugas Email

Wadah SSIS yang didukung meliputi:

- Untuk Loop Container
- Foreach Loop Kontainer

- urutan Kontainer

Mengonversi SSIS keAWS Glue StudiobersamaAWS SCT

Anda dapat menggunakanAWS SCTuntuk mengkonversi paket Microsoft SQL Server Integration Services (SSIS) keAWS Glue Studio.

SebuahPaket SSIStermasuk komponen yang diperlukan, seperti manajer koneksi, tugas, aliran kontrol, aliran data, parameter, event handler, dan variabel, untuk menjalankan ekstrak tertentu, mengubah, dan beban (ETL) tugas.AWS SCTmengkonversi paket SSIS ke format yang kompatibel denganAWS Glue Studio. Setelah Anda memigrasi database sumber Anda keAWS Cloud, Anda dapat menjalankan ini dikonversiAWS Glue Studiopekerjaan untuk melakukan tugas ETL.

Untuk mengonversi paket Microsoft SSIS keAWS Glue Studio, pastikan bahwa Anda menggunakanAWS SCTversi 1.0.661 atau lebih tinggi

Topik

- [Prasyarat](#)
- [Menambahkan paket SSIS keAWS SCTproyek](#)
- [Mengonversi paket SSIS keAWS Glue StudiobersamaAWS SCT](#)
- [MenciptakanAWS Glue Studiopekerjaan menggunakan kode yang dikonversi](#)
- [Membuat laporan penilaian untuk paket SSIS denganAWS SCT](#)
- [Komponen SSIS ituAWS SCTdapat dikonversi keAWS Glue Studio](#)

Prasyarat

Pada bagian ini, pelajari tentang tugas prasyarat untuk konversi paket SSIS keAWS Glue. Tugas-tugas ini termasuk membuat diperlukanAWSsumber daya di akun Anda.

Anda dapat menggunakanAWS Identity and Access Management(IAM) untuk menentukan kebijakan dan peran yang diperlukan untuk mengakses sumber daya yangAWS Glue Studiomenggunakan. Untuk informasi lebih lanjut, lihat[izin IAM untukAWS Glue Studiopengguna](#).

SetelahAWS SCTmengubah skrip sumber Anda menjadiAWS Glue Studio, unggah skrip yang dikonversi ke bucket Amazon S3. Pastikan Anda membuat bucket Amazon S3 ini dan pilih diAWSpengaturan profil layanan. Untuk informasi selengkapnya tentang membuat bucket S3,

lihat [Buat bucket S3 pertama Anda](#) di dalam Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Untuk memastikan bahwa AWS Glue Studio dapat terhubung ke penyimpanan data Anda, membuat konektor kustom dan koneksi. Selain itu, simpan kredensi database di AWS Secrets Manager.

Untuk membuat konektor khusus

1. Download driver JDBC untuk menyimpan data Anda. Untuk informasi lebih lanjut tentang driver JDBC yang AWS SCT menggunakan, lihat [Mengunduh driver database yang diperlukan](#).
 2. Unggah file driver ini ke bucket Amazon S3 Anda. Untuk informasi lebih lanjut, lihat [Unggah objek ke bucket Anda](#) di dalam Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
 3. Masuk ke AWS Management Console dan buka/bukalah AWS Glue Studio konsol di <https://console.aws.amazon.com/gluestudio/>.
 4. Pilih Konektor, lalu pilih Buat konektor khusus.
 5. Untuk Konektor S3 URL, pilih Jelajahi S3, dan pilih file driver JDBC yang Anda unggah ke bucket Amazon S3 Anda.
 6. Masukkan deskriptif Nama untuk konektor Anda. Misalnya, masukkan **SQLServer**.
 7. Untuk Jenis konektor, pilih JDBC.
 8. Untuk Nama kelas, masukkan nama kelas utama untuk driver JDBC Anda. Untuk SQL Server, masukkan **com.microsoft.sqlserver.jdbc.SQLServerDriver**.
 9. Untuk Basis URL JDBC, masukkan URL dasar JDBC. Sintaks URL dasar JDBC tergantung pada mesin database sumber Anda. Untuk SQL Server, gunakan format berikut: **jdbc:sqlserver://\$<host>:\$<port>;databaseName=\$<dbname>;user=\$<username>;password=\$<password>**.
- Pastikan Anda mengganti **<host>**, **<port>**, **<dbname>**, **<username>**, dan **<password>** dengan nilai-nilai Anda.
10. Untuk Pembatas parameter URL, masukkan titik koma (;).
 11. Pilih Buat konektor.

Untuk menyimpan kredensi database di AWS Secrets Manager

1. Masuk ke AWS Management Console dan buka konsol AWS Secrets Manager tersebut di <https://console.aws.amazon.com/secretsmanager/>.
2. Pilih Simpan rahasia baru.

3. PadaPilih tipe rahasiahalaman, lakukan hal berikut:
 - a. UntukTipe rahasia, pilihJenis rahasia lainnya.
 - b. UntukPasangan kunci/nilai, masukkan tombol-tombol berikut:**host**,**port**,**dbname**,**username**, dan**password**.

Selanjutnya, masukkan nilai Anda untuk kunci ini.

4. PadaKonfigurasi rahasiahalaman, masukkan deskriptifNama rahasia. Misalnya, masukkan **SQL_Server_secret**.
5. Pilih Selanjutnya. Kemudian, padaKonfigurasi rotasihalaman, pilihBerikutnyalagi.
6. PadaUlasanhalaman, tinjau detail rahasia Anda, lalu pilihMenyimpan.

Untuk membuat sambungan untuk konektor Anda

1. Masuk keAWS Management Console dan buka/bukalahAWS Glue Studiokonsol di<https://console.aws.amazon.com/gluestudio/>.
2. Pilih konektor yang ingin Anda buat koneksi, lalu pilihBuat koneksi.
3. PadaBuat koneksi halaman, masukkan deskriptifNama untuk koneksi Anda. Misalnya, masukkan **SQL-Server-connection**.
4. UntukAWS Rahasia, pilih rahasia yang Anda buatAWS Secrets Manager.
5. MengkonfigurasiOpsi jaringan, lalu pilihBuat koneksi.

Sekarang, Anda dapat membuatAWS Glue Studiopekerjaan dengan konektor kustom. Untuk informasi selengkapnya, lihat [MenciptakanAWS Glue Studiopekerjaan](#).

Menambahkan paket SSIS keAWS SCTproyek

Anda dapat menambahkan beberapa paket SSIS ke satuAWS SCTproyek.

Untuk menambahkan paket SSIS keAWS SCTproyek

1. Buat proyek baru diAWS SCTatau membuka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called "Membuat proyek"](#).
2. PilihTambahkan sumber dari menu, lalu pilihLayanan Integrasi SQL Server.
3. UntukNama koneksi, masukkan nama untuk paket SSIS Anda.AWS SCTmenampilkan nama ini di pohon di panel kiri.

4. Untuk Folder paket SSIS, masukkan path ke folder dengan paket SSIS sumber.
5. Pilih Tambahkan target dari menu, lalu pilih AWS Glue Studio.

Untuk terhubung ke AWS Glue Studio, AWS SCT menggunakan profil AWS Anda. Untuk informasi selengkapnya, lihat [Menyimpan profil layanan di AWS SCT](#).

6. Buat aturan pemetaan, yang mencakup paket SSIS sumber Anda dan AWS Glue Studio Target. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#).
7. Buat AWS Glue Studio koneksi di AWS Glue Studio konsol. Untuk informasi lebih lanjut, lihat [Membuat koneksi untuk konektor](#).
8. Pilih Manajer koneksi di pohon kiri, buka menu konteks (klik kanan), lalu pilih Konfigurasi koneksi.

AWS SCT menampilkan konfigurasi koneksi jendela.

9. Untuk setiap koneksi SSIS sumber, pilih AWS Glue Studio koneksi.

Mengonversi paket SSIS ke AWS Glue Studio bersama AWS SCT

Berikut, temukan cara mengonversi paket SSIS ke AWS Glue Studio memakai AWS SCT.

Untuk mengonversi paket SSIS ke AWS Glue Studio

1. Tambahkan paket SSIS Anda ke AWS SCT proyek. Untuk informasi selengkapnya, lihat [Menambahkan paket SSIS ke AWS SCT proyek](#).
2. Di panel kiri, perluas ETL dan SSIS node.
3. Pilih Paket, buka menu konteks (klik kanan), lalu pilih Mengonversi paket.

AWS SCT mengonversi paket SSIS yang Anda pilih ke file JSON. Objek JSON ini mewakili node dalam grafik asiklik terarah (DAG). Temukan file Anda yang telah dikonversi di Paket DAG simpul di pohon kanan.

4. Pilih Paket DAG, buka menu konteks (klik kanan), lalu pilih Simpan ke Amazon S3.

Sekarang Anda dapat menggunakan skrip ini untuk membuat pekerjaan di AWS Glue Studio.

Menciptakan AWS Glue Studio pekerjaan menggunakan kode yang dikonversi

Setelah mengonversi paket SSIS sumber Anda, Anda dapat menggunakan file JSON yang telah dikonversi untuk membuat AWS Glue Studio pekerjaan.

Untuk membuat AWS Glue Studio pekerjaan

1. Pilih Paket DAG di pohon kanan, buka menu konteks (klik kanan), lalu pilih **Mengkonfigurasi AWS Glue Studio pekerjaan**.
2. (Opsional) Terapkan paket ekstensi yang mengemulasi fungsi SSIS AWS Glue Studio.
3. Yang **Mengkonfigurasi AWS Glue Studio pekerjaan** jendela terbuka.

Lengkapi Properti pekerjaan dasar bagian:

- Nama— Masukkan nama Anda AWS Glue Studio pekerjaan.
- Nama file skrip- Masukkan nama skrip pekerjaan Anda.
- Parameter pekerjaan- Tambahkan parameter dan masukkan nilainya.

Pilih Selanjutnya.

4. Lengkapi Properti pekerjaan tingkat lanjut bagian:
 - Peran IAM- Pilih peran IAM yang digunakan untuk otorisasi AWS Glue Studio dan mengakses toko data.
 - Skrip file jalur S3- Masukkan jalur Amazon S3 ke skrip Anda yang telah dikonversi.
 - Direktori sementara— Masukkan jalur Amazon S3 ke direktori sementara untuk hasil menengah. AWS Glue Studio menggunakan direktori ini untuk membaca atau menulis ke Amazon Redshift.
 - AWS SCT secara otomatis menghasilkan jalur untuk pustaka Python. Anda dapat meninjau jalur ini di Jalur pustaka python yang dihasilkan. Anda tidak dapat mengedit jalur yang dibuat secara otomatis ini. Untuk menggunakan pustaka Python tambahan, masukkan path di Path pustaka python pengguna.
 - Path pustaka python pengguna- Masukkan jalur untuk pustaka Python pengguna tambahan. Pisahkan jalur Amazon S3 dengan koma.
 - Jalur guci tergantung- Masukkan jalur untuk dependen* . jar file. Pisahkan jalur Amazon S3 dengan koma.

- Jalur file yang direferensikan- Masukkan jalur untuk file tambahan, seperti file konfigurasi, yang diperlukan oleh skrip Anda. Pisahkan jalur Amazon S3 dengan koma.
- Tipe pekerja— Pilih G.1X atau G.2X.

Bila Anda memilih G.1X setiap pekerja memetakan ke 1 DPU (4 vCPU, memori 16 GB, dan disk 64 GB).

Bila Anda memilih G.2X setiap pekerja memetakan ke 2 DPU (8 vCPU, memori 32 GB, dan disk 128 GB).

- Jumlah pekerja yang diminta— Masukkan jumlah pekerja yang dialokasikan saat pekerjaan berjalan.
- Konkurensi maks- Masukkan jumlah maksimum berjalan bersamaan yang diizinkan untuk pekerjaan ini. Defaultnya adalah 1. AWS Glue mengembalikan kesalahan ketika ambang ini tercapai.
- Batas waktu kerja (menit)- Masukkan nilai batas waktu pada pekerjaan ETL Anda sebagai perlindungan terhadap pekerjaan pelarian. Defaultnya adalah 2.880 menit (48 jam) untuk pekerjaan batch. Jika pekerjaan melebihi batas ini, status pekerjaan yang dijalankan berubah menjadi TIMEOUT.
- Ambang batas pemberitahuan tunda (menit)— Masukkan ambang batas dalam beberapa menit sebelumnya AWS SCT mengirimkan pemberitahuan penundaan.
- Jumlah percobaan ulang— Masukkan jumlah kali (0-10) yang AWS Glue harus secara otomatis me-restart pekerjaan jika gagal. Pekerjaan yang mencapai batas waktu tidak dimulai ulang. Defaultnya adalah 0.

Pilih Selesai.

AWS SCT mengkonfigurasi pilihan Anda AWS Glue Studio pekerjaan.

5. Temukan pekerjaan Anda yang telah dikonfigurasi di bawah Lowongan kerja ETL di pohon yang tepat. Pilih pekerjaan yang dikonfigurasi, buka menu konteks (klik kanan), lalu pilih Buat AWS Glue Studio pekerjaan.
6. Pilih Terapkan status dan pastikan bahwa Status nilai untuk pekerjaan Anda Sukses.
7. Buka AWS Glue Studio konsol, pilih Segarkan, dan pilih pekerjaanmu. Kemudian pilih Jalankan.

Membuat laporan penilaian untuk paket SSIS dengan AWS SCT

Yang Laporan penilaian migrasi ETL memberikan informasi tentang mengonversi paket SSIS Anda ke format yang kompatibel dengan AWS Glue Studio. Laporan penilaian mencakup item tindakan untuk komponen paket SSIS Anda. Item tindakan ini menunjukkan komponen mana AWS SCT tidak dapat dikonversi secara otomatis.

Membuat laporan penilaian migrasi ETL

1. Perluas SSIS simpul di bawah ETL di panel kiri.
2. Pilih Paket, buka menu konteks (klik kanan), lalu pilih Buat laporan.
3. Lihat Ringkas tab. Di sini, AWS SCT menampilkan informasi ringkasan eksekutif dari laporan penilaian migrasi ETL. Ini mencakup hasil konversi untuk semua komponen paket SSIS Anda.
4. (Opsional) Simpan salinan lokal laporan penilaian migrasi ETL sebagai file PDF atau file nilai dipisahkan koma (CSV):

- Untuk menyimpan laporan penilaian migrasi ETL sebagai file PDF, pilih Simpan ke PDF di kanan atas.

File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.

- Untuk menyimpan laporan penilaian migrasi ETL sebagai file CSV, pilih Simpan ke CSV di kanan atas.

AWS SCT menciptakan tiga file CSV. File-file ini berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.

5. Pilih Item tindakan tab. Tab ini berisi daftar item yang memerlukan konversi manual AWS Glue Studio. Saat Anda memilih item tindakan dari daftar, AWS SCT menyoroti item dari paket SSIS sumber Anda bahwa item tindakan berlaku untuk.

Komponen SSIS itu AWS SCT dapat dikonversi ke AWS Glue Studio

Anda dapat menggunakan AWS SCT untuk mengkonversi komponen aliran data SSIS dan parameter ke AWS Glue Studio.

Komponen aliran data yang didukung meliputi:

- Tujuan ADO NET

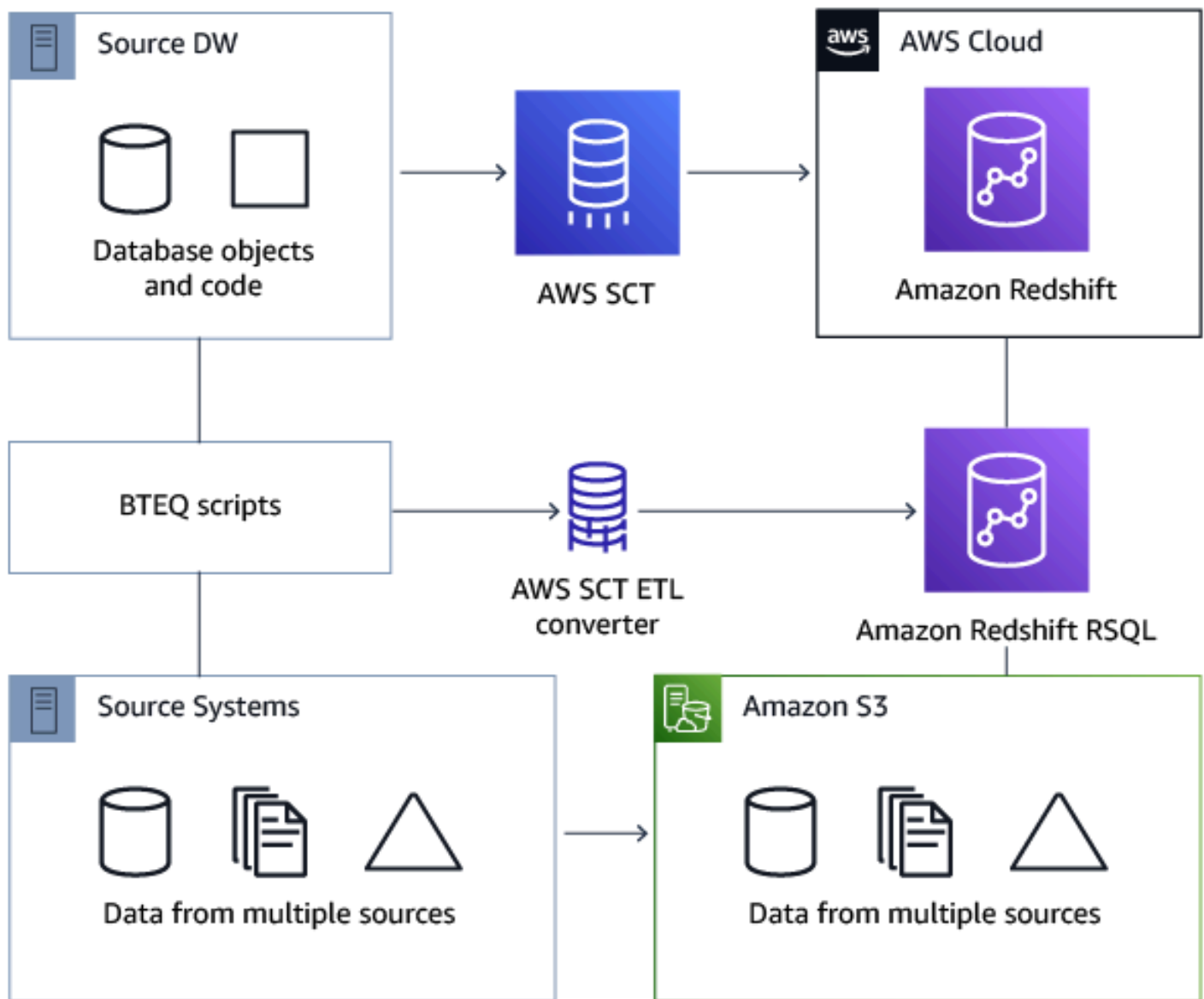
- Sumber ADO NET
- Agregat
- Peta Karakter
- Berpisah Bersyarat
- Salin Kolom
- Konversi Data
- Kolom Berasal
- Pencarian
- Gabung
- Gabung Bergabung
- Multicast
- ODBC Destination
- ODBC source
- OLEDB destinasi
- OLEDB sumber
- Hitungan Baris
- Urutkan
- Tujuan SQL Server
- Union Semua

AWS SCT dapat mengkonversi lebih banyak komponen SSIS untuk AWS Glue. Untuk informasi selengkapnya, lihat [Komponen SSIS itu AWS SCT dapat dikonversi ke AWS Glue](#).

Mengonversi skrip Teradata BTEQ ke Amazon Redshift RSQL dengan AWS SCT

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skrip Teradata Basic Teradata Query (BTEQ) ke Amazon Redshift RSQL.

Diagram arsitektur berikut menunjukkan proyek migrasi database yang menyertakan konversi skrip ekstrak, transformasi, dan pemuatan (ETL) ke Amazon Redshift RSQL.



Topik

- [Menambahkan skrip BTEQ ke AWS SCT proyek](#)
- [Mengkonfigurasi variabel substitusi dalam skrip BTEQ dengan AWS SCT](#)
- [Mengonversi skrip Teradata BTEQ ke Amazon Redshift RSQL dengan AWS SCT](#)
- [Mengelola skrip BTEQ dengan AWS SCT](#)
- [Membuat laporan penilaian konversi skrip BTEQ dengan AWS SCT](#)
- [Mengedit dan menyimpan skrip BTEQ Anda yang telah dikonversi AWS SCT](#)

Menambahkan skrip BTEQ keAWS SCTproyek

Anda dapat menambahkan beberapa skrip ke satuAWS SCTproyek.

Untuk menambahkan skrip BTEQ keAWS SCTproyek

1. Buat proyek baru diAWS SCTatau membuka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called “Membuat proyek”](#).
2. PilihTambahkan sumberdari menu, lalu pilihTeradatauntuk menambahkan database sumber Anda ke proyek. Untuk informasi selengkapnya, lihat [Menggunakan Teradata sebagai sumber](#).
3. PilihTambahkan targetdari menu untuk menambahkan database Amazon Redshift target keAWS SCTproyek.

Anda dapat menggunakan platform database target Amazon Redshift virtual. Untuk informasi selengkapnya, lihat [Menggunakan target virtual](#).

4. Buat aturan pemetaan baru yang menyertakan database Teradata sumber Anda dan target Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Menambahkan aturan pemetaan baru](#).
5. PadaLihatmenu, pilihTampilan utama.
6. Di panel kiri, perluasSkripsimpul.
7. PilihSkrip BTEQ, buka menu konteks (klik kanan), lalu pilihMemuat skrip.
8. Masukkan lokasi kode sumber untuk skrip Teradata BTEQ Anda dan pilihPilih folder.

AWS SCTmenampilkanMemuat skripjendela.

9. Lakukan salah satu dari berikut:
 - a. Jika skrip Teradata BTEQ Anda tidak menyertakan variabel substitusi, pilihTidak ada variabel substitusi, dan kemudian pilihOKEuntuk menambahkan skrip keAWS SCTproyek.
 - b. Jika skrip Teradata BTEQ Anda menyertakan variabel substitusi, konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip BTEQ](#).

Mengkonfigurasi variabel substitusi dalam skrip BTEQ denganAWS SCT

Skrip Teradata BTEQ Anda dapat menyertakan variabel substitusi. Misalnya, Anda dapat menggunakan satu skrip BTEQ dengan variabel substitusi untuk menjalankan set perintah yang

sama di beberapa lingkungan database. Anda dapat menggunakan AWS SCT untuk mengkonfigurasi variabel substitusi dalam skrip BTEQ Anda.

Sebelum Anda menjalankan skrip BTEQ dengan variabel substitusi, pastikan untuk menetapkan nilai untuk semua variabel. Untuk melakukan ini, Anda dapat menggunakan alat atau aplikasi lain seperti skrip Bash, UC4 (Automic), dan sebagainya. AWS SCT dapat menyelesaikan dan mengkonversi variabel substitusi hanya setelah Anda menetapkan nilai-nilai mereka.

Untuk mengkonfigurasi variabel substitusi dalam skrip BTEQ Anda

1. Tambahkan skrip BTEQ Anda ke AWS SCT proyek. Untuk informasi selengkapnya, lihat [Menambahkan skrip BTEQ ke AWS SCT proyek](#).

Saat Anda menambahkan skrip, pilih variabel substitusi digunakan.

2. Untuk tentukan format variabel, masukkan ekspresi reguler yang cocok dengan semua variabel substitusi dalam skrip Anda.

Misalnya, jika nama variabel substitusi Anda dimulai dengan `{` dan diakhiri dengan `}`, gunakan `\${w+}` ekspresi reguler. Untuk mencocokkan variabel substitusi yang dimulai dengan tanda dolar atau tanda persen, gunakan `\$w+ | \%w+` ekspresi reguler.

Ekspresi reguler sesuai dengan sintaks ekspresi reguler Java dalam templat AWS SCT Anda. Untuk informasi lebih lanjut, lihat [java.util.regex Pola Kelas](#) dalam dokumentasi Java.

3. Pilih `OK` untuk memuat skrip ke AWS SCT proyek, dan kemudian pilih `OK` untuk menutup Memuat skrip jendela.
4. Pilih variabel untuk melihat semua variabel substitusi ditemukan dan nilai-nilai mereka.
5. Untuk Nilai, masukkan nilai untuk variabel substitusi.

Mengonversi skrip Teradata BTEQ ke Amazon Redshift RSQL dengan AWS SCT

Setelah itu, temukan cara mengonversi skrip BTEQ ETL ke Amazon Redshift RSQL menggunakan AWS SCT.

Untuk mengonversi skrip Teradata BTEQ ke Amazon Redshift RSQL

1. Tambahkan skrip BTEQ Anda ke AWS SCT proyek. Untuk informasi selengkapnya, lihat [Menambahkan skrip BTEQ ke AWS SCT proyek](#).

2. Konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip BTEQ](#).
 3. Di panel kiri, perluasSkripsimpul.
 4. Lakukan salah satu dari berikut:
 - Untuk mengonversi skrip BTEQ tunggal, perluasSkrip BTEQnode, pilih skrip yang akan dikonversi, lalu pilihKonversi ke RSQLdari menu konteks (klik kanan).
 - Untuk menyembunyikan beberapa skrip, pastikan Anda memilih semua skrip untuk dikonversi. Kemudian pilihSkrip BTEQ, buka menu konteks (klik kanan), lalu pilihKonversi ke RSQLdi bawahKonversi skrip.
- AWS SCTmengubah semua skrip Teradata BTEQ pilihan Anda ke format yang kompatibel dengan Amazon Redshift RSQL. Temukan skrip Anda yang telah dikonversi diSkripnode di panel database target.
5. Edit skrip Amazon Redshift RSQL yang telah dikonversi, atau simpan. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan skrip BTEQ Anda yang telah dikonversi](#).

Mengelola skrip BTEQ denganAWS SCT

Anda dapat menambahkan beberapa skrip BTEQ atau menghapus skrip BTEQ dariAWS SCTproyek.

Untuk menambahkan skrip BTEQ tambahan keAWS SCTproyek

1. PerluasSkripsimpul di panel kiri.
2. PilihSkrip BTEQsimpul, dan buka menu konteks (klik kanan).
3. PilihMemuat skrip.
4. Masukkan informasi yang diperlukan untuk menambahkan skrip BTEQ baru dan konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Menambahkan skrip BTEQ keAWS SCTproyek](#) dan [Mengkonfigurasi variabel substitusi dalam skrip BTEQ](#).

Untuk menghapus skrip BTEQ dariAWS SCTproyek

1. PerluasSkrip BTEQsimpul di bawahSkripdi panel kiri.
2. Pilih skrip yang akan dihapus, dan buka menu konteks (klik kanan).
3. PilihHapus skrip.

Membuat laporan penilaian konversi skrip BTEQ dengan AWS SCT

SEBUAH laporan penilaian konversi skrip BTEQ memberikan informasi tentang mengonversi perintah BTEQ dan pernyataan SQL dari skrip BTEQ Anda ke format yang kompatibel dengan Amazon Redshift RSQL. Laporan penilaian mencakup item tindakan untuk perintah BTEQ dan pernyataan SQL yang AWS SCT tidak dapat mengkonversi.

Untuk membuat laporan penilaian konversi skrip BTEQ

1. Perluas Skrip BTEQ simpul di bawah Skrip di panel kiri.
2. Pilih skrip yang akan dikonversi, dan buka menu konteks (klik kanan).
3. Pilih Konversi ke RSQL di bawah Buat laporan.
4. Lihat Ringkasan tab. Yang Ringkasan tab menampilkan informasi ringkasan eksekutif dari laporan penilaian skrip BTEQ. Ini termasuk hasil konversi untuk semua perintah BTEQ dan pernyataan SQL dari skrip BTEQ Anda.
5. (Opsional) Simpan salinan lokal dari laporan penilaian konversi skrip BTEQ sebagai file PDF atau file nilai dipisahkan koma (CSV):
 - Untuk menyimpan laporan penilaian konversi skrip BTEQ sebagai file PDF, pilih Simpan ke PDF di kanan atas.

File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.
 - Untuk menyimpan laporan penilaian konversi skrip BTEQ sebagai file CSV, pilih Simpan ke CSV di kanan atas.

File CSV berisi item tindakan, tindakan yang disarankan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.
6. Pilih Item tindak tab. Tab ini berisi daftar item yang memerlukan konversi manual ke Amazon Redshift RSQL. Saat Anda memilih item tindakan dari daftar, AWS SCT menyoroti item dari skrip BTEQ sumber Anda yang berlaku untuk item tindakan.

Mengedit dan menyimpan skrip BTEQ Anda yang telah dikonversi AWS SCT

Anda dapat mengedit skrip yang telah dikonversi di panel bawah AWS SCT proyek. AWS SCT menyimpan skrip yang diedit sebagai bagian dari proyek Anda.

Untuk menyimpan skrip yang telah dikonversi

1. PerluasSkrip RSQLsimpul di bawahSkripdi panel database target.
2. Pilih skrip yang telah dikonversi, buka menu konteks (klik kanan), dan pilihSimpan skrip.
3. Masukkan path ke folder untuk menyimpan skrip yang dikonversi dan pilihSimpan.

AWS SCTmenyimpan skrip yang dikonversi ke file dan membuka file ini.

Mengonversi skrip shell dengan perintah Teradata BTEQ yang disematkan ke Amazon Redshift RSQL denganAWS SCT

Anda dapat menggunakanAWS Schema Conversion Tool(AWS SCT) untuk mengonversi skrip shell dengan perintah Teradata Basic Teradata Query (BTEQ) yang disematkan ke skrip shell dengan perintah Amazon Redshift RSQL yang disematkan.

AWS SCTmengeksktrah perintah Teradata BTEQ dari skrip shell Anda dan mengubahnya menjadi format yang kompatibel dengan Amazon Redshift. Setelah Anda memigrasi database Teradata ke Amazon Redshift, Anda dapat menggunakan skrip yang dikonversi ini untuk mengelola database Amazon Redshift baru Anda.

Anda juga dapat menggunakanAWS SCTuntuk mengonversi file dengan skrip ETL Teradata BTEQ ke Amazon Redshift RSQL. Untuk informasi selengkapnya, lihat [Mengonversi skrip Teradata BTEQ ke Amazon Redshift RSQL denganAWS SCT](#).

Topik

- [Menambahkan skrip shell dengan perintah Teradata BTEQ yang disematkan keAWS SCTproyek](#)
- [Mengkonfigurasi variabel substitusi dalam skrip shell dengan perintah Teradata BTEQ tertanam denganAWS SCT](#)
- [Mengkonversi skrip shell dengan perintah Teradata BTEQ tertanam denganAWS SCT](#)
- [Mengelola skrip shell dengan perintah Teradata BTEQ tertanam denganAWS SCT](#)
- [Membuat laporan penilaian untuk konversi skrip shell denganAWS SCT](#)
- [Mengedit dan menyimpan skrip shell Anda yang telah dikonversiAWS SCT](#)

Menambahkan skrip shell dengan perintah Teradata BTEQ yang disematkan keAWS SCTproyek

Anda dapat menambahkan beberapa skrip ke satuAWS SCTproyek.

Untuk menambahkan skrip shell keAWS SCTproyek

1. Buat proyek baru diAWS SCTatau membuka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called “Membuat proyek”](#).
2. PilihTambahkan sumberdari menu, lalu pilihTeradatauntuk menambahkan database sumber Anda ke proyek. Untuk informasi selengkapnya, lihat [Menggunakan Teradata sebagai sumber](#).
3. PilihTambahkan targetdari menu dan untuk menambahkan database Amazon Redshift target keAWS SCTproyek.

Anda dapat menggunakan platform database target Amazon Redshift virtual. Untuk informasi selengkapnya, lihat [Menggunakan target virtual](#).

4. Buat aturan pemetaan baru yang menyertakan database Teradata sumber Anda dan target Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Menambahkan aturan pemetaan baru](#).
5. PadaLihatmenu, pilihTampilan utama.
6. Di panel kiri, perluasSkripsimpul.
7. PilihCangkang, buka menu konteks (klik kanan), lalu pilihMemuat skrip.
8. Masukkan lokasi skrip shell sumber Anda dengan perintah Teradata BTEQ yang disematkan dan pilihPilih folder.

AWS SCTmenampilkanMemuat skripjendela.

9. Lakukan salah satu dari berikut:
 - Jika skrip shell Anda tidak menyertakan variabel substitusi, pilihTidak ada variabel substitusi, dan kemudian pilihOKEuntuk menambahkan skrip keAWS SCTproyek.
 - Jika skrip shell Anda menyertakan variabel substitusi, konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip shell](#).

Mengkonfigurasi variabel substitusi dalam skrip shell dengan perintah Teradata BTEQ tertanam denganAWS SCT

Skrip shell Anda dapat menyertakan variabel substitusi. Misalnya, Anda dapat menggunakan satu skrip dengan variabel substitusi untuk mengelola database di lingkungan yang berbeda. Anda dapat menggunakanAWS SCTuntuk mengkonfigurasi variabel substitusi dalam skrip shell Anda.

Sebelum Anda menjalankan perintah BTEQ dengan variabel substitusi dari skrip shell, pastikan untuk menetapkan nilai untuk semua variabel di dalam skrip shell ini.AWS SCTdapat menyelesaikan dan mengkonversi variabel substitusi hanya setelah Anda menetapkan nilai-nilai mereka.

Untuk mengkonfigurasi variabel substitusi dalam skrip shell Anda

1. Tambahkan skrip shell sumber Anda keAWS SCTproyek. Untuk informasi selengkapnya, lihat [Menambahkan skrip shell keAWS SCTproyek](#).

Saat Anda menambahkan skrip, pilihVariabel substitusi digunakan.

2. UntukTentukan format variabel, masukkan ekspresi reguler yang cocok dengan semua variabel substitusi dalam skrip Anda.

Misalnya, jika nama variabel substitusi Anda dimulai dengan\${dan diakhiri dengan}, gunakan`\${\w+}`ekspresi reguler. Untuk mencocokkan variabel substitusi yang dimulai dengan tanda dolar atau tanda persen, gunakan`\$\w+|\%\w+`ekspresi reguler.

Ekspresi reguler sesuai dengan sintaks ekspresi reguler Java dalam templat AWS SCT Anda. Untuk informasi lebih lanjut, lihat[java.util.regex Pola Kelas](#)dalam dokumentasi Java.

3. PilihOKEuntuk memuat skrip keAWS SCTproyek, dan kemudian pilihOKEuntuk menutupMemuat skripjendela.
4. PilihVariabeluntuk melihat semua variabel substitusi ditemukan dan nilai-nilai mereka.
5. UntukNilai, masukkan nilai untuk variabel substitusi.

Mengkonversi skrip shell dengan perintah Teradata BTEQ tertanam denganAWS SCT

Setelah itu, temukan cara mengonversi skrip shell dengan perintah Teradata BTEQ yang disematkan ke skrip shell dengan perintah Amazon Redshift RSQL yang disematkan menggunakanAWS SCT.

Untuk mengonversi skrip shell

1. Tambahkan skrip shell Anda keAWS SCTproyek. Untuk informasi selengkapnya, lihat [Menambahkan skrip shell keAWS SCTproyek](#).
2. Konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip shell](#).
3. Di panel kiri, perluasSkripsimpul.
4. Lakukan salah satu dari berikut:
 - Untuk mengonversi perintah BTEQ dari skrip shell tunggal, perluasCangkangnode, pilih skrip yang akan dikonversi, lalu pilihKonversi skripdari menu konteks (klik kanan).
 - Untuk menyembunyikan beberapa skrip, pastikan Anda memilih semua skrip untuk dikonversi. Kemudian pilihCangkang, buka menu konteks (klik kanan), lalu pilihKonversi skrip.
5. Pilih OKE.

AWS SCTmengubah perintah BTEQ dalam skrip shell pilihan Anda ke format yang kompatibel dengan Amazon Redshift RSQL. Temukan skrip Anda yang telah dikonversi diSkripnode di panel database target.

6. Edit skrip Amazon Redshift RSQL yang telah dikonversi atau simpan. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan skrip shell yang telah dikonversi](#).

Mengelola skrip shell dengan perintah Teradata BTEQ tertanam denganAWS SCT

Anda dapat menambahkan beberapa skrip shell atau menghapus skrip shell dariAWS SCTproyek.

Untuk menambahkan skrip shell baru keAWS SCTproyek

1. PerluasSkripsimpul di panel kiri.
2. PilihCangkangsimpul, dan buka menu konteks (klik kanan).
3. PilihMemuat skrip.
4. Masukkan informasi yang diperlukan untuk menambahkan skrip shell baru dan mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Menambahkan skrip shell keAWS SCTproyek](#) dan [Mengkonfigurasi variabel substitusi dalam skrip shell](#).

Untuk menghapus skrip shell dari AWS SCT proyek

1. Perluas Cangkang simpul di bawah Skrip di panel kiri.
2. Pilih skrip yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus skrip.

Membuat laporan penilaian untuk konversi skrip shell dengan AWS SCT

Yang laporan penilaian konversi skrip shell memberikan informasi tentang mengubah perintah BTEQ dan pernyataan SQL. Konversi dilakukan dari skrip sumber Anda ke format yang kompatibel dengan Amazon Redshift RSQL. Laporan penilaian mencakup item tindakan untuk perintah BTEQ dan pernyataan SQL yang AWS SCT tidak dapat mengkonversi.

Membuat laporan penilaian konversi skrip shell

1. Perluas Cangkang simpul di bawah Skrip di panel kiri.
2. Pilih skrip yang akan dikonversi, buka menu konteks (klik kanan), lalu pilih Buat laporan.
3. Lihat Ringkas tab. Yang Ringkas tab menampilkan informasi ringkasan eksekutif dari laporan penilaian skrip shell. Ini termasuk hasil konversi untuk semua perintah BTEQ dan pernyataan SQL dari skrip sumber Anda.
4. (Opsional) Simpan salinan lokal dari laporan penilaian konversi skrip shell sebagai file PDF atau file nilai dipisahkan koma (CSV):
 - Untuk menyimpan laporan penilaian konversi skrip shell sebagai file PDF, pilih Simpan ke PDF di kanan atas.

File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.
 - Untuk menyimpan laporan penilaian konversi skrip shell sebagai file CSV, pilih Simpan ke CSV di kanan atas.

File CSV berisi item tindakan, tindakan yang disarankan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.
5. Pilih Item tindak tab. Tab ini berisi daftar item yang memerlukan konversi manual ke Amazon Redshift RSQL. Saat Anda memilih item tindakan dari daftar, AWS SCT menyoroti item dari skrip shell sumber Anda yang berlaku untuk item tindakan.

Mengedit dan menyimpan skrip shell Anda yang telah dikonversi AWS SCT

Anda dapat mengedit skrip yang telah dikonversi di panel bawah AWS SCT proyek AWS SCT menyimpan skrip yang diedit sebagai bagian dari proyek Anda.

Untuk menyimpan skrip yang telah dikonversi

1. Perluas Skrip RSQL simpul di bawah Skrip di panel database target.
2. Pilih skrip yang telah dikonversi, buka menu konteks (klik kanan), dan pilih Simpan skrip.
3. Masukkan path ke folder untuk menyimpan skrip yang dikonversi dan pilih Simpan.

AWS SCT menyimpan skrip yang dikonversi ke file dan membuka file ini.

Konversi Teradata FastExport skrip pekerjaan untuk Amazon Redshift RSQL dengan AWS SCT

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengkonversi Teradata FastExport skrip pekerjaan ke Amazon Redshift RSQL.

SEBUAH FastExport skrip pekerjaan adalah satu set FastExport perintah dan pernyataan SQL yang memilih dan mengekspor data dari database Teradata. AWS SCT berubah FastExport perintah dan pernyataan SQL ke format yang kompatibel dengan Amazon Redshift RSQL. Setelah memigrasi database Teradata ke Amazon Redshift, Anda dapat menggunakan skrip yang dikonversi ini untuk mengekspor data dari database Amazon Redshift.

Topik

- [Menambahkan FastExport skrip pekerjaan untuk AWS SCT proyek](#)
- [Mengkonfigurasi variabel substitusi di Teradata FastExport skrip pekerjaan dengan AWS SCT](#)
- [Konversi Teradata FastExport skrip pekerjaan dengan AWS SCT](#)
- [Mengelola Teradata FastExport skrip pekerjaan dengan AWS SCT](#)
- [Membuat laporan penilaian untuk Teradata FastExport konversi skrip pekerjaan dengan AWS SCT](#)
- [Mengedit dan menyimpan Teradata Anda yang telah dikonversi FastExport skrip pekerjaan dengan AWS SCT](#)

Menambahkan FastExportskrip pekerjaan untuk AWS SCT proyek

Anda dapat menambahkan beberapa skrip ke satu AWS SCT proyek.

Untuk menambahkan FastExportskrip pekerjaan untuk Anda AWS SCT proyek

1. Buat proyek baru di AWS SCT atau membuka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called “Membuat proyek”](#).
2. Pilih Tambahkan sumber dari menu, lalu pilih Teradata untuk menambahkan database sumber Anda ke proyek. Untuk informasi selengkapnya, lihat [Menggunakan Teradata sebagai sumber](#).
3. Pilih Tambahkan target dari menu dan untuk menambahkan database Amazon Redshift target ke AWS SCT proyek.

Anda dapat menggunakan platform database target Amazon Redshift virtual. Untuk informasi selengkapnya, lihat [Menggunakan target virtual](#).

4. Buat aturan pemetaan baru yang menyertakan database Teradata sumber Anda dan target Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Menambahkan aturan pemetaan baru](#).
5. Pada Lihat menu, pilih Tampilan utama.
6. Di panel kiri, perluas Skrip simpul.
7. Pilih FastExport, buka menu konteks (klik kanan), lalu pilih Muat skrip.
8. Masukkan lokasi kode sumber untuk Teradata Anda FastExportskrip pekerjaan dan pilih folder.

AWS SCT menampilkan Muat skrip jendela.

9. Lakukan salah satu dari berikut:
 - Jika Anda Teradata FastExportskrip pekerjaan tidak menyertakan variabel substitusi, pilih Tidak ada variabel substitusi dan kemudian pilih OKE untuk menambahkan skrip ke AWS SCT proyek.
 - Jika Anda Teradata FastExportskrip pekerjaan termasuk variabel substitusi, mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi di FastExportskrip pekerjaan](#).

Mengkonfigurasi variabel substitusi di TeradataFastExportskrip pekerjaan denganAWS SCT

Teradata AndaFastExportskrip pekerjaan dapat mencakup variabel substitusi. Misalnya, Anda dapat menggunakan skrip tunggal dengan variabel substitusi untuk mengeksport data dari beberapa database. Anda dapat menggunakanAWS SCTuntuk mengkonfigurasi variabel substitusi dalam skrip Teradata Anda.

Sebelum Anda menjalankanFastExportskrip pekerjaan dengan variabel substitusi, pastikan untuk menetapkan nilai-nilai untuk semua variabel. Untuk melakukan ini, Anda dapat menggunakan alat atau aplikasi lain seperti skrip Bash, UC4 (Automic), dan sebagainya.AWS SCTdapat menyelesaikan dan mengkonversi variabel substitusi hanya setelah Anda menetapkan nilai-nilai mereka.

Untuk mengkonfigurasi variabel substitusi diFastExportskrip pekerjaan

1. Tambahkan sumber Anda TeradataFastExportskrip pekerjaan untuk AndaAWS SCTproyek. Untuk informasi selengkapnya, lihat [Menambahkan skrip BTEQ keAWS SCTproyek](#).

Saat Anda menambahkan skrip, pilihVariabel substitusi digunakan.

2. UntukTentukan format variabel, masukkan ekspresi reguler yang cocok dengan semua variabel substitusi dalam skrip Anda.

Misalnya, jika nama variabel substitusi Anda dimulai dengan\${dan diakhiri dengan}, gunakan\\${\w+}ekspresi reguler. Untuk mencocokkan variabel substitusi yang dimulai dengan tanda dolar atau tanda persen, gunakan\\$\w+|\%\w+ekspresi reguler.

Ekspresi reguler sesuai dengan sintaks ekspresi reguler Java dalam templat AWS SCT Anda. Untuk informasi lebih lanjut, lihat[java.util.regex Pola Kelas](#)dalam dokumentasi Java.

3. PilihOKEuntuk memuat skrip keAWS SCTproyek, dan kemudian pilihOKEuntuk menutupMemuat skripjendela.
4. Di panel kiri, perluasSkripsimpul. PilihFastExport, dan kemudian pilih folder Anda dengan skrip. Buka menu konteks (klik kanan), lalu pilihVariabel ekspordi bawahVariabel substitusi.
5. Variabel substitusi ekspor untuk satu skrip. Perluas folder Anda dengan skrip, pilih skrip Anda, buka menu konteks (klik kanan), dan pilihVariabel ekspordi bawahVariabel substitusi.
6. Masukkan nama file nilai dipisahkan koma (CSV) untuk menyimpan variabel substitusi dan pilihSimpan.
7. Buka file CSV ini dan isi nilai untuk variabel substitusi.

Tergantung pada sistem operasi, AWS SCT menggunakan format yang berbeda untuk file CSV. Nilai-nilai dalam file mungkin baik tertutup dalam tanda kutip atau tidak. Pastikan bahwa Anda menggunakan format yang sama untuk nilai-nilai variabel substitusi sebagai nilai-nilai lain dalam file. AWS SCT tidak dapat mengimpor file CSV dengan nilai dalam format yang berbeda.

8. Simpan file CSV.
9. Di panel kiri, perluas Skrip simpul. Pilih FastExport, dan kemudian pilih skrip Anda. Buka menu konteks (klik kanan), lalu pilih Variabel impordi bawah Variabel substitusi.
10. Pilih file CSV Anda, lalu pilih Terbuka.
11. Pilih Variabel untuk melihat semua variabel substitusi ditemukan dan nilai-nilai mereka.

Konversi Teradata FastExport skrip pekerjaan dengan AWS SCT

Berikut, temukan cara mengonversi Teradata FastExport pekerjaan untuk Amazon Redshift RSQL menggunakan AWS SCT.

Untuk mengonversi Teradata FastExport skrip pekerjaan untuk Amazon Redshift RSQL

1. Tambahkan FastExport skrip pekerjaan untuk Anda AWS SCT proyek. Untuk informasi selengkapnya, lihat [Menambahkan FastExport skrip pekerjaan untuk AWS SCT proyek](#).
2. Konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi di FastExport skrip pekerjaan](#).
3. Di panel kiri, perluas Skrip simpul.
4. Lakukan salah satu dari berikut:
 - Untuk mengonversi satu FastExport skrip pekerjaan, perluas FastExport node, pilih skrip yang akan dikonversi, lalu pilih Konversi skrip dari menu konteks (klik kanan).
 - Untuk menyembunyikan beberapa skrip, pastikan Anda memilih semua skrip untuk dikonversi. Kemudian pilih FastExport, buka menu konteks (klik kanan), lalu pilih Konversi skrip.

AWS SCT mengkonversi semua Teradata pilihan Anda FastExport skrip pekerjaan ke format yang kompatibel dengan Amazon Redshift RSQL. Temukan skrip Anda yang telah dikonversi di Skrip node di panel database target.

5. Edit skrip Amazon Redshift RSQL yang telah dikonversi atau simpan. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan konversi Anda FastExport skrip pekerjaan](#).

Mengelola TeradataFastExportskrip pekerjaan denganAWS SCT

Anda dapat menambahkan beberapa TeradataFastExportskrip pekerjaan atau menghapusFastExportnaskah pekerjaan dari AndaAWS SCTproyek.

Untuk menambahkan yang baruFastExportskrip pekerjaan untuk AndaAWS SCTproyek

1. PerluasSkripsimpul di panel kiri.
2. PilihFastExportsimpul, dan buka menu konteks (klik kanan).
3. PilihMemuat skrip.
4. Masukkan informasi yang diperlukan untuk menambahkan yang baruFastExportscript pekerjaan dan mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [MenambahkanFastExportskrip pekerjaan untukAWS SCTproyek](#) dan [Mengkonfigurasi variabel substitusi diFastExportskrip pekerjaan](#).

Untuk menghapusFastExportnaskah pekerjaan dari AndaAWS SCTproyek

1. PerluasFastExportsimpul di bawahSkripdi panel kiri.
2. Pilih skrip yang akan dihapus, dan buka menu konteks (klik kanan).
3. PilihHapus skrip.

Membuat laporan penilaian untuk TeradataFastExportkonversi skrip pekerjaan denganAWS SCT

YangFastExportlaporan penilaian konversi skrip pekerjaanmemberikan informasi tentang mengkonversiFastExportperintah dan pernyataan SQL dariFastExportskrip ke format yang kompatibel dengan Amazon Redshift RSQL. Laporan penilaian mencakup item tindakan untukFastExportperintah dan pernyataan SQL yangAWS SCTTidak dapat mengkonversi.

Membuat laporan penilaian konversi skrip untuk TeradataFastExportpekerjaan

1. PerluasFastExportsimpul di bawahSkripdi panel kiri.
2. Pilih skrip yang akan dikonversi, buka menu konteks (klik kanan), lalu pilihBuat laporan.
3. LihatRingasantab. YangRingasantab menampilkan informasi ringkasan eksekutif dariFastExportlaporan penilaian skrip pekerjaan. Ini mencakup hasil konversi untuk semuaFastExportperintah dan pernyataan SQL dari skrip sumber Anda.

4. Anda dapat menyimpan salinan lokalFastExportlaporan penilaian konversi skrip pekerjaan baik sebagai file PDF atau file nilai dipisahkan koma (CSV).
 - a. Untuk menyimpanFastExportlaporan penilaian konversi skrip pekerjaan sebagai file PDF, pilihSimpan ke PDFdi kanan atas.

File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.
 - b. Untuk menyimpanFastExportlaporan penilaian konversi skrip pekerjaan sebagai file CSV, pilihSimpan ke CSVdi kanan atas.

File CSV berisi item tindakan, tindakan yang disarankan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.
5. PilihItem tindakantab. Tab ini berisi daftar item yang memerlukan konversi manual ke Amazon Redshift RSQL. Saat Anda memilih item tindakan dari daftar,AWS SCTmenyoroti item dari sumber AndaFastExportskrip pekerjaan yang item tindakan berlaku untuk.

Mengedit dan menyimpan Teradata Anda yang telah dikonversiFastExportskrip pekerjaan denganAWS SCT

Anda dapat mengedit skrip yang telah dikonversi di panel bawahAWS SCTproyek.AWS SCTmenyimpan skrip yang diedit sebagai bagian dari proyek Anda.

Untuk menyimpan skrip yang telah dikonversi

1. PerluasSkrip RSQLsimpul di bawahSkripdi panel database target.
2. Pilih skrip yang telah dikonversi, buka menu konteks (klik kanan), dan pilihSimpan skrip.
3. Masukkan path ke folder untuk menyimpan skrip yang dikonversi dan pilihSimpan.

AWS SCTmenyimpan skrip yang dikonversi ke file dan membuka file ini.

Konversi TeradataFastLoadskrip pekerjaan untuk Amazon Redshift RSQL denganAWS SCT

Anda dapat menggunakanAWS Schema Conversion Tool(AWS SCT) untuk mengkonversi TeradataFastLoadskrip pekerjaan ke Amazon Redshift RSQL.

SEBUAH Teradata FastLoad skrip adalah satu set perintah yang menggunakan beberapa sesi untuk memuat data dalam tabel kosong pada Database Teradata. Teradata FastLoad memproses serangkaian Teradata FastLoad perintah dan pernyataan SQL. Teradata FastLoad perintah memberikan kontrol sesi dan penanganan data transfer data. Pernyataan SQL membuat, memelihara, dan menjatuhkan tabel.

AWS SCT mengkonversi Teradata FastLoad perintah dan pernyataan SQL ke format yang kompatibel dengan Amazon Redshift RSQL. Setelah memigrasi database Teradata ke Amazon Redshift, Anda dapat menggunakan skrip yang dikonversi ini untuk memuat data ke database Amazon Redshift Anda.

Topik

- [Menambahkan FastLoad skrip pekerjaan untuk Anda AWS SCT proyek](#)
- [Mengkonfigurasi variabel substitusi di Teradata FastLoad skrip pekerjaan dengan AWS SCT](#)
- [Konversi Teradata FastLoad skrip pekerjaan dengan AWS SCT](#)
- [Mengelola Teradata FastLoad skrip pekerjaan dengan AWS SCT](#)
- [Membuat laporan penilaian untuk Teradata FastLoad konversi skrip pekerjaan dengan AWS SCT](#)
- [Mengedit dan menyimpan Teradata Anda yang telah dikonversi FastLoad skrip pekerjaan dengan AWS SCT](#)

Menambahkan FastLoad skrip pekerjaan untuk Anda AWS SCT proyek

Anda dapat menambahkan beberapa skrip ke satu AWS SCT proyek.

Untuk menambahkan FastLoad skrip pekerjaan untuk Anda AWS SCT proyek

1. Buat proyek baru di AWS SCT, atau membuka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called "Membuat proyek"](#).
2. Pilih **Tambahkan sumber** dari menu, lalu pilih **Teradata** untuk menambahkan database sumber Anda ke proyek. Untuk informasi selengkapnya, lihat [Menggunakan Teradata sebagai sumber](#).
3. Pilih **Tambahkan target** dari menu dan tambahkan database Amazon Redshift target ke AWS SCT proyek.

Anda dapat menggunakan platform database target Amazon Redshift virtual. Untuk informasi selengkapnya, lihat [Menggunakan target virtual](#).

4. Buat aturan pemetaan baru yang menyertakan database Teradata sumber Anda dan target Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Menambahkan aturan pemetaan baru](#).
5. Pada Lihat menu, pilih Tampilan utama.
6. Di panel kiri, perluas Skrip simpul.
7. Pilih FastLoad, buka menu konteks (klik kanan), lalu pilih Muat skrip.
8. Masukkan lokasi sumber Anda Teradata FastLoad skrip pekerjaan dan pilih folder.

AWS SCT menampilkan Muat skrip jendela.
9. Lakukan salah satu dari berikut:
 - Jika Anda Teradata FastLoad skrip pekerjaan tidak menyertakan variabel substitusi, pilih Tidak ada variabel substitusi, dan kemudian pilih OK untuk menambahkan skrip ke AWS SCT proyek.
 - Jika Anda Teradata FastLoad skrip pekerjaan termasuk variabel substitusi, mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi di FastLoad skrip pekerjaan](#).

Mengkonfigurasi variabel substitusi di Teradata FastLoad skrip pekerjaan dengan AWS SCT

Teradata Anda FastLoad skrip pekerjaan mungkin termasuk variabel substitusi. Misalnya, Anda dapat menggunakan skrip tunggal dengan variabel substitusi untuk memuat data ke database yang berbeda.

Sebelum Anda menjalankan FastLoad skrip pekerjaan dengan variabel substitusi, pastikan untuk menetapkan nilai-nilai untuk semua variabel. Untuk melakukan ini, Anda dapat menggunakan alat atau aplikasi lain seperti skrip Bash, UC4 (Automic), dan sebagainya.

AWS SCT dapat menyelesaikan dan mengkonversi variabel substitusi hanya setelah Anda menetapkan nilai-nilai mereka. Sebelum Anda memulai konversi sumber Anda Teradata FastLoad skrip pekerjaan, pastikan bahwa Anda menetapkan nilai untuk semua variabel substitusi. Anda dapat menggunakan AWS SCT untuk mengkonfigurasi variabel substitusi dalam skrip Teradata Anda.

Untuk mengkonfigurasi variabel substitusi diFastLoadskrip pekerjaan

1. Ketika Anda menambahkan sumber Anda TeradataFastLoadskrip pekerjaan untuk AndaAWS SCTproyek, pilihVariabel substitusi digunakan. Untuk informasi selengkapnya tentang menambahkan skrip ini, lihat[MenambahkanFastLoadskrip pekerjaan untuk AndaAWS SCTproyek](#).
2. UntukTentukan format variabel, masukkan ekspresi reguler yang cocok dengan semua variabel substitusi dalam skrip Anda.

Misalnya, jika nama variabel substitusi Anda dimulai dengan\${dan diakhiri dengan}, gunakan\\${\w+}ekspresi reguler. Untuk mencocokkan variabel substitusi yang dimulai dengan tanda dolar atau tanda persen, gunakan\\$\w+|\%\w+ekspresi reguler.

Ekspresi reguler sesuai dengan sintaks ekspresi reguler Java dalam templat AWS SCT Anda. Untuk informasi lebih lanjut, lihat[java.util.regex Pola Kelas](#)dalam dokumentasi Java.

3. PilihOKEuntuk memuat skrip keAWS SCTproyek, dan kemudian pilihOKEuntuk menutupMemuat skripjendela.
4. Di panel kiri, perluasSkripsimpul. PilihFastLoad, dan kemudian pilih folder Anda dengan skrip. Buka menu konteks (klik kanan), lalu pilihVariabel ekspordi bawahVariabel substitusi.

Selain itu, Anda dapat mengeksport variabel substitusi untuk satu skrip. Perluas folder Anda dengan skrip, pilih skrip Anda, buka menu konteks (klik kanan), dan pilihVariabel ekspordi bawahVariabel substitusi.

5. Masukkan nama file nilai dipisahkan koma (CSV) untuk menyimpan variabel substitusi, dan kemudian pilihSimpan.
6. Buka file CSV ini dan isi nilai untuk variabel substitusi.

Tergantung pada sistem operasi,AWS SCTmenggunakan format yang berbeda untuk file CSV. Nilai-nilai dalam file mungkin baik tertutup dalam tanda kutip atau tidak. Pastikan bahwa Anda menggunakan format yang sama untuk nilai-nilai variabel substitusi sebagai nilai-nilai lain dalam file.AWS SCTtidak dapat mengimpor file CSV dengan nilai dalam format yang berbeda.

7. Simpan file CSV.
8. Di panel kiri, perluasSkripsimpul. PilihFastLoad, dan kemudian pilih skrip Anda. Buka menu konteks (klik kanan), lalu pilihVariabel impordi bawahVariabel substitusi.
9. Pilih file CSV Anda, lalu pilihTerbuka.
10. PilihVariabeluntuk melihat semua variabel substitusi ditemukan dan nilai-nilai mereka.

Konversi TeradataFastLoadskrip pekerjaan denganAWS SCT

Berikut, temukan cara mengonversi TeradataFastLoadpekerjaan untuk Amazon Redshift RSQL menggunakanAWS SCT.

Untuk mengonversi TeradataFastLoadskrip pekerjaan untuk Amazon Redshift RSQL

1. TambahkanFastLoadskrip pekerjaan untuk AndaAWS SCTproyek. Untuk informasi selengkapnya, lihat [MenambahkanFastLoadskrip pekerjaan untuk AndaAWS SCTproyek](#).
2. Konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi diFastLoadskrip pekerjaan](#).
3. Di panel kiri, perluasSkripsimpul.
4. Lakukan salah satu dari berikut:

- Untuk mengonversi satuFastLoadskrip pekerjaan, perluasFastLoadnode, pilih skrip yang akan dikonversi, lalu pilihKonversi skripdari menu konteks (klik kanan).
- Untuk menyembunyikan beberapa skrip, pastikan Anda memilih semua skrip untuk dikonversi. PilihFastLoad, buka menu konteks (klik kanan), lalu pilihKonversi skrip. Kemudian lakukan salah satu dari berikut ini:
 - Jika Anda menyimpan file data sumber di Amazon S3, pilihjalur objek S3untukLokasi file data sumber.

Masukkan nilai untukFolder bucket Amazon S3danBucket Amazon S3 untuk file manifesuntuk file data sumber Anda.

- Jika Anda tidak menyimpan file data sumber di Amazon S3, pilihAlamat hostuntukLokasi file data sumber.

Masukkan nilai untukURL atau alamat IP host,Nama login pengguna host, danBucket Amazon S3 untuk file manifesuntuk file data sumber Anda.

5. Pilih OKE.

AWS SCTmengkonversi semua Teradata pilihan AndaFastLoadskrip pekerjaan ke format yang kompatibel dengan Amazon Redshift RSQL. Temukan skrip Anda yang telah dikonversi diSkripnode di panel database target.

6. Edit skrip Amazon Redshift RSQL yang telah dikonversi atau simpan. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan konversi AndaFastLoadskrip pekerjaan](#).

Mengelola TeradataFastLoadskrip pekerjaan denganAWS SCT

Anda dapat menambahkan beberapa TeradataFastLoadskrip pekerjaan atau menghapusFastLoadnaskah pekerjaan dari AndaAWS SCTproyek.

Untuk menambahkan yang baruFastLoadskrip pekerjaan untuk AndaAWS SCTproyek

1. PerluasSkripsimpul di panel kiri.
2. PilihFastLoadnode dan membuka konteks (klik kanan) menu.
3. PilihMemuat skrip.
4. Masukkan informasi yang diperlukan untuk menambahkan yang baruFastLoadscript pekerjaan dan mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [MenambahkanFastLoadskrip pekerjaan untuk AndaAWS SCTproyek](#) dan [Mengkonfigurasi variabel substitusi diFastLoadskrip pekerjaan](#).

Untuk menghapusFastLoadnaskah pekerjaan dari AndaAWS SCTproyek

1. PerluasFastLoadsimpul di bawahSkripdi panel kiri.
2. Pilih skrip yang akan dihapus, dan buka menu konteks (klik kanan).
3. PilihHapus skrip.

Membuat laporan penilaian untuk TeradataFastLoadkonversi skrip pekerjaan denganAWS SCT

YangFastLoadlaporan penilaian konversi skrip pekerjaanmemberikan informasi tentang mengkonversiFastLoadperintah dan pernyataan SQL. Konversi dilakukan dari skrip sumber Anda ke format yang kompatibel dengan Amazon Redshift RSQL. Laporan penilaian mencakup item tindakan untukFastLoadperintah dan pernyataan SQL yangAWS SCTTidak dapat mengkonversi.

Membuat laporan penilaian konversi skrip untuk TeradataFastLoadpekerjaan

1. PerluasFastLoadsimpul di bawahSkripdi panel kiri.
2. Pilih skrip yang akan dikonversi, buka menu konteks (klik kanan), lalu pilihBuat laporan.
3. LihatRingkasantab.

Yang Ringkas tab menampilkan informasi ringkasan eksekutif dari FastLoad laporan penilaian skrip pekerjaan. Ini mencakup hasil konversi untuk semua FastLoad perintah dan pernyataan SQL dari skrip sumber Anda.

4. (Opsional) Simpan salinan lokal FastLoad laporan penilaian konversi skrip pekerjaan baik sebagai file PDF atau file nilai dipisahkan koma (CSV):

- Untuk menyimpan FastLoad laporan penilaian konversi skrip pekerjaan sebagai file PDF, pilih Simpan ke PDF di kanan atas.

File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.

- Untuk menyimpan FastLoad laporan penilaian konversi skrip pekerjaan sebagai file CSV, pilih Simpan ke CSV di kanan atas.

File CSV berisi item tindakan, tindakan yang disarankan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.

5. Pilih Item tindakan tab. Tab ini berisi daftar item yang memerlukan konversi manual ke Amazon Redshift RSQL. Saat Anda memilih item tindakan dari daftar, AWS SCT menyoroti item dari sumber Anda FastLoad skrip pekerjaan yang item tindakan berlaku untuk.

Mengedit dan menyimpan Teradata Anda yang telah dikonversi FastLoad skrip pekerjaan dengan AWS SCT

Anda dapat mengedit skrip yang telah dikonversi di panel bawah AWS SCT proyek. AWS SCT menyimpan skrip yang diedit sebagai bagian dari proyek Anda.

Untuk menyimpan skrip yang telah dikonversi

1. Perluas Skrip RSQL simpul di bawah Skrip di panel database target.
2. Pilih skrip yang telah dikonversi, buka menu konteks (klik kanan), dan pilih Simpan skrip.
3. Masukkan path ke folder untuk menyimpan skrip yang dikonversi dan pilih Simpan.

AWS SCT menyimpan skrip yang dikonversi ke file dan membuka file ini.

Konversi TeradataMultiLoadskrip pekerjaan untuk Amazon Redshift RSQL denganAWS SCT

Anda dapat menggunakanAWS SCTuntuk mengkonversi TeradataMultiLoadskrip pekerjaan ke Amazon Redshift RSQL.

SEBUAHTeradataMultiLoadskrip pekerjaanadalah seperangkat perintah untuk pemeliharaan batch Database Teradata Anda. TeradataMultiLoadtugas impor melakukan sejumlah insert yang berbeda, update, dan menghapus operasi pada hingga lima tabel yang berbeda dan pandangan. TeradataMultiLoadmenghapus tugas dapat menghapus sejumlah besar baris dari satu tabel.

AWS SCTmengkonversi TeradataMultiLoadperintah dan pernyataan SQL ke format yang kompatibel dengan Amazon Redshift RSQL. Setelah Anda memigrasi database Teradata ke Amazon Redshift, gunakan skrip yang dikonversi ini untuk mengelola data dalam database Amazon Redshift Anda.

Topik

- [MenambahkanMultiLoadskrip pekerjaan untuk AndaAWS SCTproyek](#)
- [Mengkonfigurasi variabel substitusi di TeradataMultiLoadskrip pekerjaan denganAWS SCT](#)
- [Konversi TeradataMultiLoadskrip pekerjaan denganAWS SCT](#)
- [Mengelola TeradataMultiLoadskrip pekerjaan denganAWS SCT](#)
- [Membuat laporan penilaian untuk TeradataMultiLoadkonversi skrip pekerjaan denganAWS SCT](#)
- [Mengedit dan menyimpan Teradata Anda yang telah dikonversiMultiLoadskrip pekerjaan denganAWS SCT](#)

MenambahkanMultiLoadskrip pekerjaan untuk AndaAWS SCTproyek

Anda dapat menambahkan beberapa skrip ke satuAWS SCTproyek.

Untuk menambahkanMultiLoadskrip pekerjaan untuk AndaAWS SCTproyek

1. Buat proyek baru diAWS SCTatau membuka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called “Membuat proyek”](#).
2. PilihTambahkan sumberdari menu, lalu pilihTeradatauntuk menambahkan database sumber Anda ke proyek. Untuk informasi selengkapnya, lihat [Menggunakan Teradata sebagai sumber](#).
3. PilihTambahkan targetdari menu dan untuk menambahkan database Amazon Redshift target keAWS SCTproyek.

Anda dapat menggunakan platform database target Amazon Redshift virtual. Untuk informasi selengkapnya, lihat [Menggunakan target virtual](#).

4. Buat aturan pemetaan baru yang menyertakan database Teradata sumber Anda dan target Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Menambahkan aturan pemetaan baru](#).
5. Pada Lihat menu, pilih Tampilan utama.
6. Di panel kiri, perluas Skrip simpul.
7. Pilih MultiLoad, buka menu konteks (klik kanan), lalu pilih Muat skrip.
8. Masukkan lokasi sumber Anda Teradata MultiLoad skrip pekerjaan dan pilih folder.

AWS SCT menampilkan Muat skrip jendela.

9. Lakukan salah satu dari berikut:
 - Jika Anda Teradata MultiLoad skrip pekerjaan tidak menyertakan variabel substitusi, pilih Tidak ada variabel substitusi, dan kemudian pilih OKE untuk menambahkan skrip ke AWS SCT proyek.
 - Jika Anda Teradata MultiLoad skrip pekerjaan termasuk variabel substitusi, mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi di MultiLoad skrip pekerjaan](#).

Mengkonfigurasi variabel substitusi di Teradata MultiLoad skrip pekerjaan dengan AWS SCT

Teradata Anda MultiLoad skrip pekerjaan mungkin termasuk variabel substitusi. Misalnya, Anda dapat menggunakan skrip tunggal dengan variabel substitusi untuk memuat data ke database yang berbeda.

Sebelum Anda menjalankan MultiLoad skrip pekerjaan dengan variabel substitusi, pastikan untuk menetapkan nilai-nilai untuk semua variabel. Untuk melakukan ini, Anda dapat menggunakan alat atau aplikasi lain seperti skrip Bash, UC4 (Automatic), dan sebagainya.

AWS SCT dapat menyelesaikan dan mengkonversi variabel substitusi hanya setelah Anda menetapkan nilai-nilai mereka. Sebelum Anda memulai konversi sumber Anda Teradata MultiLoad skrip pekerjaan, pastikan bahwa Anda menetapkan nilai untuk semua variabel substitusi. Anda dapat menggunakan AWS SCT untuk mengkonfigurasi variabel substitusi dalam skrip Teradata Anda.

Untuk mengkonfigurasi variabel substitusi diMultiLoadskrip pekerjaan

1. Ketika Anda menambahkan sumber Anda TeradataMultiLoadskrip pekerjaan untuk AndaAWS SCTproyek, pilihVariabel substitusi digunakan. Untuk informasi selengkapnya tentang menambahkan skrip ini, lihat[MenambahkanMultiLoadskrip pekerjaan untuk AndaAWS SCTproyek](#).
2. UntukTentukan format variabel, masukkan ekspresi reguler yang cocok dengan semua variabel substitusi dalam skrip Anda.

Misalnya, jika nama variabel substitusi Anda dimulai dengan\${dan diakhiri dengan}, gunakan\\${\w+}ekspresi reguler. Untuk mencocokkan variabel substitusi yang dimulai dengan tanda dolar atau tanda persen, gunakan\\$\w+|\%\w+ekspresi reguler.

Ekspresi reguler sesuai dengan sintaks ekspresi reguler Java dalam templat AWS SCT Anda. Untuk informasi lebih lanjut, lihat[java.util.regex Pola Kelas](#)dalam dokumentasi Java.

3. PilihOKEuntuk memuat skrip keAWS SCTproyek, dan kemudian pilihOKEuntuk menutupMemuat skripjendela.
4. PilihVariabeluntuk melihat semua variabel substitusi ditemukan dan nilai-nilai mereka.
5. UntukNilai, masukkan nilai untuk variabel substitusi.

Konversi TeradataMultiLoadskrip pekerjaan denganAWS SCT

Berikut, temukan cara mengonversi TeradataMultiLoadpekerjaan untuk Amazon Redshift RSQL menggunakanAWS SCT.

Untuk mengonversi TeradataMultiLoadskrip pekerjaan untuk Amazon Redshift RSQL

1. TambahkanMultiLoadskrip pekerjaan untuk AndaAWS SCTproyek. Untuk informasi selengkapnya, lihat [MenambahkanMultiLoadskrip pekerjaan untuk AndaAWS SCTproyek](#).
2. Konfigurasi variabel substitusi dan masukkan nilainya. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi diMultiLoadskrip pekerjaan](#).
3. Di panel kiri, perluasSkripsimpul.
4. Lakukan salah satu dari berikut:
 - Untuk mengonversi satuMultiLoadskrip pekerjaan, perluasMultiLoadnode, pilih skrip yang akan dikonversi, lalu pilihKonversi skripdari menu konteks (klik kanan).

- Untuk menyembunyikan beberapa skrip, pastikan Anda memilih semua skrip untuk dikonversi. Pilih **MultiLoad**, buka menu konteks (klik kanan), lalu pilih **Konversi skrip**.
5. Lakukan salah satu dari berikut:
- Jika Anda menyimpan file data sumber di Amazon S3, pilih jalur objek S3 untuk lokasi file data sumber.

Memasukkan folder bucket Amazon S3 dan Bucket Amazon S3 untuk file manifest untuk file data sumber Anda.
 - Jika Anda tidak menyimpan file data sumber di Amazon S3, pilih alamat host untuk lokasi file data sumber.

Memasukkan URL atau alamat IP host, Nama login pengguna host, dan Bucket Amazon S3 untuk file manifest untuk file data sumber Anda.
6. Pilih **OKE**.
- AWS SCT mengkonversi semua Teradata pilihan Anda **MultiLoad** skrip pekerjaan ke format yang kompatibel dengan Amazon Redshift RSQL. Temukan skrip Anda yang telah dikonversi di **Skripnode** di panel database target.
7. Edit skrip Amazon Redshift RSQL yang telah dikonversi atau simpan. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan konversi Anda **MultiLoad** skrip pekerjaan](#).

Mengelola Teradata **MultiLoad** skrip pekerjaan dengan AWS SCT

Anda dapat menambahkan beberapa Teradata **MultiLoad** skrip pekerjaan atau menghapus **MultiLoad** skrip pekerjaan dari Anda AWS SCT proyek.

Untuk menambahkan yang baru **MultiLoad** skrip pekerjaan untuk Anda AWS SCT proyek

1. Perluas **Skrip** simpul di panel kiri.
2. Pilih **MultiLoad** node dan membuka konteks (klik kanan) menu.
3. Pilih **Memuat skrip**.
4. Masukkan informasi yang diperlukan untuk menambahkan yang baru **MultiLoad** skrip pekerjaan dan mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Menambahkan **MultiLoad** skrip pekerjaan untuk Anda AWS SCT proyek](#) dan [Mengkonfigurasi variabel substitusi di **MultiLoad** skrip pekerjaan](#).

Untuk menghapus MultiLoad naskah pekerjaan dari Anda AWS SCT proyek

1. Perluas MultiLoad simpul di bawah Skrip di panel kiri.
2. Pilih skrip yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus skrip.

Membuat laporan penilaian untuk Teradata MultiLoad konversi skrip pekerjaan dengan AWS SCT

Yang MultiLoad laporan penilaian konversi skrip pekerjaan memberikan informasi tentang mengkonversi MultiLoad perintah dan pernyataan SQL. Konversi dilakukan dari skrip sumber Anda ke perintah Amazon Redshift RSQL dan pernyataan SQL untuk Amazon Redshift. Laporan penilaian mencakup item tindakan untuk MultiLoad perintah dan pernyataan SQL yang AWS SCT tidak dapat mengkonversi.

Membuat laporan penilaian konversi skrip untuk Teradata MultiLoad pekerjaan

1. Perluas MultiLoad simpul di bawah Skrip di panel kiri.
2. Pilih skrip untuk membuat laporan penilaian, buka menu konteks (klik kanan), lalu pilih Buat laporan.
3. Lihat Ringkas tab. Yang Ringkas tab menampilkan informasi ringkasan eksekutif dari MultiLoad laporan penilaian skrip pekerjaan. Ini mencakup hasil konversi untuk semua MultiLoad perintah dan pernyataan SQL dari skrip sumber Anda.
4. (Opsional) Simpan salinan lokal MultiLoad laporan penilaian konversi skrip pekerjaan sebagai file PDF atau file nilai dipisahkan koma (CSV):

- Untuk menyimpan MultiLoad laporan penilaian konversi skrip pekerjaan sebagai file PDF, pilih Simpan ke PDF di kanan atas.

File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.

- Untuk menyimpan MultiLoad laporan penilaian konversi skrip pekerjaan sebagai file CSV, pilih Simpan ke CSV di kanan atas.

AWS SCT menciptakan dua file CSV. File-file ini berisi ringkasan eksekutif, item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.

5. Pilih item tindakan tab. Tab ini berisi daftar item yang memerlukan konversi manual ke Amazon Redshift RSQL. Saat Anda memilih item tindakan dari daftar, AWS SCT menyoroti item dari sumber Anda MultiLoad skrip pekerjaan yang item tindakan berlaku untuk.

Mengedit dan menyimpan Teradata Anda yang telah dikonversi MultiLoad skrip pekerjaan dengan AWS SCT

Anda dapat mengedit skrip yang telah dikonversi di panel bawah AWS SCT proyek. AWS SCT menyimpan skrip yang diedit sebagai bagian dari proyek Anda.

Untuk menyimpan skrip yang telah dikonversi

1. Perluas skrip RSQL simpul di bawah skrip di panel database target.
2. Pilih skrip yang telah dikonversi, buka menu konteks (klik kanan), dan pilih Simpan skrip.
3. Masukkan path ke folder untuk menyimpan skrip yang dikonversi dan pilih Simpan.

AWS SCT menyimpan skrip yang dikonversi ke file dan membuka file ini.

Memigrasi kerangka kerja data besar dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk memigrasi kerangka data besar ke AWS Cloud.

Saat ini, AWS SCT mendukung migrasi kluster Hadoop ke Amazon EMR dan Amazon S3. Proses migrasi ini mencakup layanan Hive dan HDFS.

Selain itu, Anda dapat menggunakannya AWS SCT untuk mengotomatiskan konversi alur kerja orkestrasi Apache Oozie Anda. AWS Step Functions

Topik

- [Memigrasi Apache Hadoop ke Amazon EMR dengan AWS Schema Conversion Tool](#)
- [Mengkonversi Apache Oozie ke dengan AWS Step FunctionsAWS Schema Conversion Tool](#)

Memigrasi Apache Hadoop ke Amazon EMR dengan AWS Schema Conversion Tool

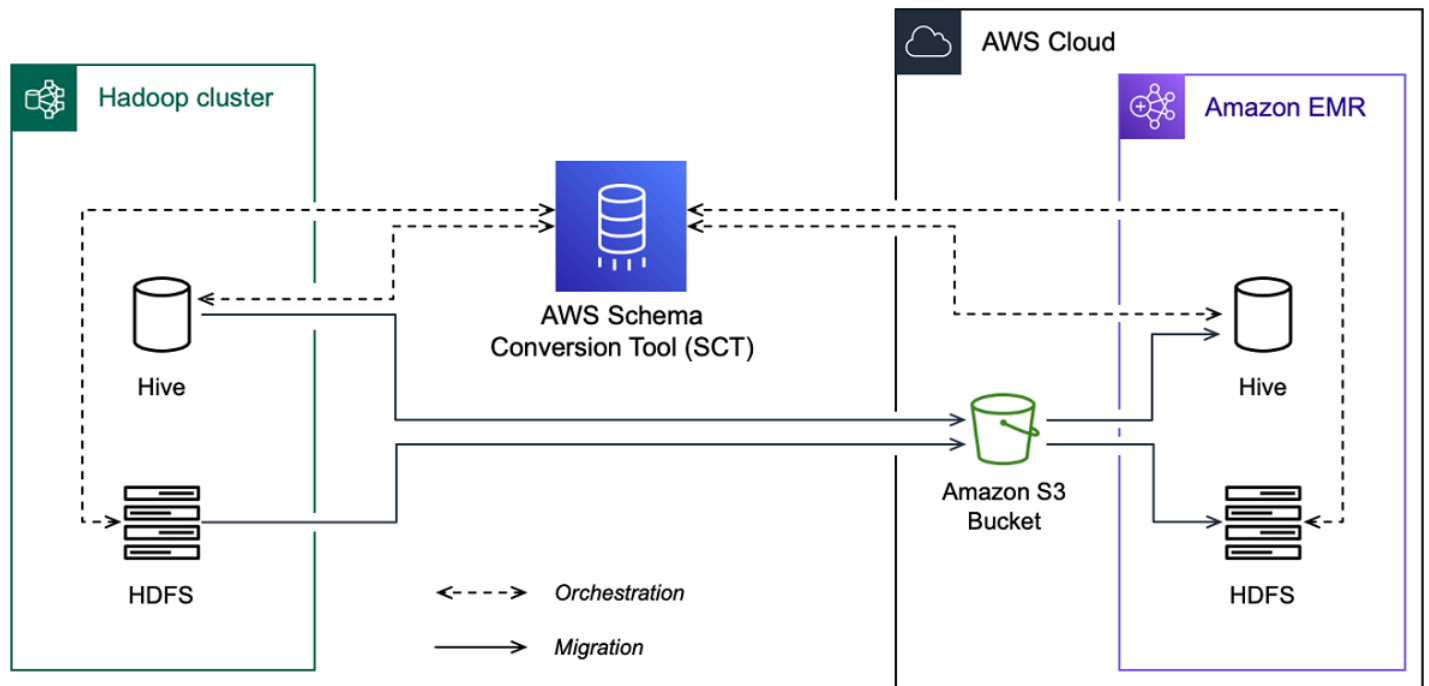
Untuk memigrasi kluster Apache Hadoop, pastikan Anda menggunakan AWS SCT versi 1.0.670 atau yang lebih tinggi. Juga, membiasakan diri dengan antarmuka baris perintah (CLI) dari AWS SCT. Untuk informasi selengkapnya, lihat [AWS SCT Referensi CLI](#).

Topik

- [Ikhtisar migrasi](#)
- [Langkah 1: Hubungkan ke kluster Hadoop Anda](#)
- [Langkah 2: Mengatur aturan pemetaan](#)
- [Langkah 3: Buat laporan penilaian](#)
- [Langkah 4: Migrasikan kluster Apache Hadoop Anda ke Amazon EMR dengan AWS SCT](#)
- [Menjalankan skrip CLI Anda](#)
- [Mengelola proyek migrasi data besar](#)

Ikhtisar migrasi

Gambar berikut menunjukkan diagram arsitektur migrasi dari Apache Hadoop ke Amazon EMR.



AWS SCT memigrasi data dan metadata dari kluster Hadoop sumber Anda ke bucket Amazon S3. Selanjutnya, AWS SCT gunakan metadata sumber Hive Anda untuk membuat objek database di layanan Amazon EMR Hive target. Opsional, Anda dapat mengkonfigurasi Hive untuk menggunakan AWS Glue Data Catalog sebagai metastore nya. Dalam kasus ini, AWS SCT migrasi metadata sumber Hive Anda ke AWS Glue Data Catalog

Kemudian, Anda dapat menggunakan AWS SCT untuk memigrasi data dari bucket Amazon S3 ke layanan Amazon EMR HDFS target Anda. Atau, Anda dapat meninggalkan data di bucket Amazon S3 Anda dan menggunakannya sebagai repositori data untuk beban kerja Hadoop Anda.

Untuk memulai migrasi Hadoop, Anda membuat dan menjalankan skrip AWS SCT CLI Anda. Script ini mencakup set lengkap perintah untuk menjalankan migrasi. Anda dapat men-download dan mengedit template dari script migrasi Hadoop. Untuk informasi selengkapnya, lihat [Mendapatkan skenario CLI](#).

Pastikan skrip Anda menyertakan langkah-langkah berikut sehingga Anda dapat menjalankan migrasi dari Apache Hadoop ke Amazon S3 dan Amazon EMR.

Langkah 1: Hubungkan ke kluster Hadoop Anda

Untuk memulai migrasi cluster Apache Hadoop Anda, buat proyek baru. AWS SCT Selanjutnya, sambungkan ke kluster sumber dan target Anda. Pastikan Anda membuat dan menyediakan AWS sumber daya target sebelum memulai migrasi.

Pada langkah ini, Anda menggunakan perintah AWS SCT CLI berikut.

- `CreateProject`— untuk membuat AWS SCT proyek baru.
- `AddSourceCluster`— untuk terhubung ke sumber Hadoop cluster dalam proyek Anda. AWS SCT
- `AddSourceClusterHive`— untuk terhubung ke layanan sumber Hive dalam proyek Anda.
- `AddSourceClusterHDFS`— untuk terhubung ke layanan HDFS sumber dalam proyek Anda.
- `AddTargetCluster`— untuk terhubung ke kluster Amazon EMR target dalam proyek Anda.
- `AddTargetClusterS3`— untuk menambahkan bucket Amazon S3 ke proyek Anda.
- `AddTargetClusterHive`— untuk terhubung ke layanan sarang target dalam proyek Anda
- `AddTargetClusterHDFS`— untuk terhubung ke layanan HDFS target dalam proyek Anda

Untuk contoh menggunakan perintah AWS SCT CLI ini, lihat [Menggunakan Apache Hadoop sebagai sumber](#).

Ketika Anda menjalankan perintah yang menghubungkan ke cluster sumber atau target, AWS SCT mencoba untuk membuat koneksi ke cluster ini. Jika upaya koneksi gagal, maka AWS SCT berhenti menjalankan perintah dari skrip CLI Anda dan menampilkan pesan kesalahan.

Langkah 2: Mengatur aturan pemetaan

Setelah Anda terhubung ke kluster sumber dan target, siapkan aturan pemetaan. Aturan pemetaan mendefinisikan target migrasi untuk kluster sumber. Pastikan Anda menyiapkan aturan pemetaan untuk semua kluster sumber yang Anda tambahkan dalam proyek Anda AWS SCT. Untuk informasi selengkapnya tentang aturan pemetaan, lihat [Membuat aturan pemetaan di AWS SCT](#).

Pada langkah ini, Anda menggunakan `AddServerMapping` perintah. Perintah ini menggunakan dua parameter, yang menentukan cluster sumber dan target. Anda dapat menggunakan `AddServerMapping` perintah dengan jalur eksplisit ke objek database Anda atau dengan nama objek. Untuk opsi pertama, Anda menyertakan jenis objek dan namanya. Untuk opsi kedua, Anda hanya menyertakan nama objek.

- `sourceTreePath`- jalur eksplisit ke objek database sumber Anda.
`targetTreePath`- jalur eksplisit ke objek database target Anda.
- `sourceNamePath`- jalur yang hanya mencakup nama-nama objek sumber Anda.
`targetNamePath`- jalur yang hanya mencakup nama objek target Anda.

Contoh kode berikut membuat aturan pemetaan menggunakan jalur eksplisit untuk sumber `testdb` Hive database dan target EMR cluster.

```
AddServerMapping
-sourceTreePath: 'Clusters.HADOOP_SOURCE.HIVE_SOURCE.Databases.testdb'
-targetTreePath: 'Clusters.HADOOP_TARGET.HIVE_TARGET'
/
```

Anda dapat menggunakan contoh ini dan contoh berikut di Windows. Untuk menjalankan perintah CLI di Linux, pastikan Anda memperbarui jalur file dengan tepat untuk sistem operasi Anda.

Contoh kode berikut membuat aturan pemetaan menggunakan jalur yang hanya mencakup nama objek.

```
AddServerMapping
-sourceNamePath: 'HADOOP_SOURCE.HIVE_SOURCE.testdb'
-targetNamePath: 'HADOOP_TARGET.HIVE_TARGET'
/
```

Anda dapat memilih Amazon EMR atau Amazon S3 sebagai target untuk objek sumber Anda. Untuk setiap objek sumber, Anda dapat memilih hanya satu target dalam satu AWS SCT proyek. Untuk mengubah target migrasi objek sumber, hapus aturan pemetaan yang ada, lalu buat aturan pemetaan baru. Untuk menghapus aturan pemetaan, gunakan `DeleteServerMapping` perintah. Perintah ini menggunakan salah satu dari dua parameter berikut.

- `sourceTreePath`- jalur eksplisit ke objek database sumber Anda.
- `sourceNamePath`- jalur yang hanya mencakup nama-nama objek sumber Anda.

Untuk informasi lebih lanjut tentang `AddServerMapping` dan `DeleteServerMapping` perintah, lihat [Referensi AWS Schema Conversion Tool CLI](#).

Langkah 3: Buat laporan penilaian

Sebelum memulai migrasi, sebaiknya buat laporan penilaian. Laporan ini merangkum semua tugas migrasi dan merinci item tindakan yang akan muncul selama migrasi. Untuk memastikan migrasi Anda tidak gagal, lihat laporan ini dan alamat item tindakan sebelum migrasi. Untuk informasi selengkapnya, lihat [Laporan penilaian migrasi](#).

Pada langkah ini, Anda menggunakan `CreateMigrationReport` perintah. Perintah ini menggunakan dua parameter. `treePath` parameter-nya wajib, dan `forceMigrate` parameter-nya opsional.

- `treePath`— jalur eksplisit ke objek database sumber tempat Anda menyimpan salinan laporan penilaian.
- `forceMigrate`- ketika diatur ke `true`, AWS SCT melanjutkan migrasi bahkan jika proyek Anda menyertakan folder HDFS dan tabel Hive yang merujuk ke objek yang sama. Nilai default-nya adalah `false`.

Anda kemudian dapat menyimpan salinan laporan penilaian sebagai file PDF atau nilai dipisahkan koma (CSV). Untuk melakukannya, gunakan `SaveReportCSV` perintah `SaveReportPDF` or.

`SaveReportPDF`Perintah menyimpan salinan laporan penilaian Anda sebagai file PDF. Perintah ini menggunakan empat parameter. `file` parameter-nya wajib, parameter lainnya bersifat opsional.

- `file`- jalur ke file PDF dan namanya.
- `filter`- nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.
- `treePath`— jalur eksplisit ke objek database sumber tempat Anda menyimpan salinan laporan penilaian.
- `namePath`— jalur yang hanya menyertakan nama objek target yang Anda simpan salinan laporan penilaian.

`SaveReportCSV`Perintah menyimpan laporan penilaian Anda dalam tiga file CSV. Perintah ini menggunakan empat parameter. `directory` parameter-nya wajib, parameter lainnya bersifat opsional.

- `directory`- path ke folder tempat AWS SCT menyimpan file CSV.

- `filter`- nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.
- `treePath`— jalur eksplisit ke objek database sumber tempat Anda menyimpan salinan laporan penilaian.
- `namePath`— jalur yang hanya menyertakan nama objek target yang Anda simpan salinan laporan penilaian.

Contoh kode berikut menyimpan salinan laporan penilaian dalam `c:\sct\ar.pdf` file.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

Contoh kode berikut menyimpan salinan laporan penilaian sebagai file CSV di `c:\sct` folder.

```
SaveReportCSV
-file:'c:\sct'
/
```

Untuk informasi lebih lanjut tentang `SaveReportPDF` dan `SaveReportCSV` perintah, lihat [Referensi AWS Schema Conversion Tool CLI](#).

Langkah 4: Migrasikan kluster Apache Hadoop Anda ke Amazon EMR dengan AWS SCT

Setelah Anda mengkonfigurasi AWS SCT proyek Anda, mulai migrasi kluster Apache Hadoop lokal Anda ke. AWS Cloud

Pada langkah ini, Anda menggunakan `Migrate`, `MigrationStatus`, dan `ResumeMigration` perintah.

`Migrate`Perintah memigrasikan objek sumber Anda ke cluster target. Perintah ini menggunakan empat parameter. Pastikan bahwa Anda menentukan `filter` atau `treePath` parameter. Parameter lainnya bersifat opsional.

- `filter`- nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.

- `treePath`— jalur eksplisit ke objek database sumber tempat Anda menyimpan salinan laporan penilaian.
- `forceLoad`- ketika diatur ke `true`, AWS SCT secara otomatis memuat pohon metadata database selama migrasi. Nilai default-nya adalah `false`.
- `forceMigrate`- ketika diatur ke `true`, AWS SCT melanjutkan migrasi bahkan jika proyek Anda menyertakan folder HDFS dan tabel Hive yang merujuk ke objek yang sama. Nilai default-nya adalah `false`.

`MigrationStatusPerintah` mengembalikan informasi tentang kemajuan migrasi. Untuk menjalankan perintah ini, masukkan nama proyek migrasi Anda untuk `name` parameter. Anda menentukan nama ini dalam `CreateProject` perintah.

`ResumeMigrationPerintah` melanjutkan migrasi terputus yang Anda luncurkan menggunakan perintah. `Migrate ResumeMigrationPerintah` tidak menggunakan parameter. Untuk melanjutkan migrasi, Anda harus terhubung ke kluster sumber dan target Anda. Untuk informasi selengkapnya, lihat [Mengelola proyek migrasi](#).

Contoh kode berikut memigrasi data dari layanan HDFS sumber Anda ke Amazon EMR.

```
Migrate
  -treePath: 'Clusters.HADOOP_SOURCE.HDFS_SOURCE'
  -forceMigrate: 'true'
/
```

Menjalankan skrip CLI Anda

Setelah Anda selesai mengedit skrip AWS SCT CLI Anda, simpan sebagai file dengan `.scts` ekstensi. Sekarang, Anda dapat menjalankan skrip Anda dari `app` folder jalur AWS SCT instalasi Anda. Untuk melakukannya, gunakan perintah berikut.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\hadoop.scts"
```

Pada contoh sebelumnya, ganti *script_path dengan path* ke file Anda dengan skrip CLI. Untuk informasi selengkapnya tentang menjalankan skrip CLI AWS SCT, lihat [Mode skrip](#).

Mengelola proyek migrasi data besar

Setelah menyelesaikan migrasi, Anda dapat menyimpan dan mengedit AWS SCT proyek Anda untuk penggunaan di masa mendatang.

Untuk menyimpan AWS SCT proyek Anda, gunakan `SaveProject` perintah. Perintah ini tidak menggunakan parameter.

Contoh kode berikut menyimpan AWS SCT proyek Anda.

```
SaveProject  
/
```

Untuk membuka AWS SCT proyek Anda, gunakan `OpenProject` perintah. Perintah ini menggunakan satu parameter wajib. Untuk `file` parameter, masukkan path ke file AWS SCT proyek Anda dan namanya. Anda menentukan nama proyek dalam `CreateProject` perintah. Pastikan Anda menambahkan `.scts` ekstensi ke nama file proyek Anda untuk menjalankan `OpenProject` perintah.

Contoh kode berikut membuka `hadoop_emr` proyek dari `c:\sct` folder.

```
OpenProject  
-file: 'c:\sct\hadoop_emr.scts'  
/
```

Setelah Anda membuka AWS SCT proyek Anda, Anda tidak perlu menambahkan kluster sumber dan target karena Anda telah menemukannya ke proyek Anda. Untuk mulai bekerja dengan kluster sumber dan target Anda, Anda harus terhubung dengannya. Untuk melakukannya, Anda menggunakan `ConnectSourceCluster` dan `ConnectTargetCluster` perintah. Perintah ini menggunakan parameter yang sama dengan `AddTargetCluster` perintah `AddSourceCluster` dan. Anda dapat mengedit skrip CLI Anda dan mengganti nama perintah ini meninggalkan daftar parameter tanpa perubahan.

Contoh kode berikut menghubungkan ke sumber Hadoop cluster.

```
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'  
/  

```

Contoh kode berikut terhubung ke kluster Amazon EMR target.

```
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'  
/  

```

Dalam contoh sebelumnya, ganti *hadoop_address* dengan *alamat* IP cluster Hadoop Anda. Jika diperlukan, konfigurasi nilai variabel port. Selanjutnya, ganti *hadoop_user* dan *hadoop_password* dengan *nama pengguna Hadoop Anda dan kata sandi* untuk pengguna ini. Untuk *path\ name*, masukkan nama dan path ke file PEM untuk cluster Hadoop sumber Anda. Untuk informasi selengkapnya tentang menambahkan kluster sumber dan target Anda, lihat [Menggunakan Apache Hadoop sebagai sumber untuk AWS SCT](#).

Setelah Anda terhubung ke sumber dan menargetkan kluster Hadoop, Anda harus terhubung ke layanan Hive dan HDFS Anda, serta ke bucket Amazon S3 Anda. Untuk melakukannya, Anda menggunakan `ConnectSourceClusterHive`, `ConnectSourceClusterHdfs`, `ConnectTargetClusterHive`, `ConnectTargetClusterHdfs`, dan `ConnectTargetClusterS3` perintah. Perintah ini menggunakan parameter yang sama dengan perintah yang Anda gunakan untuk menambahkan layanan Hive dan HDFS, dan bucket Amazon S3 ke proyek Anda. Edit skrip CLI untuk mengganti `Add` awalan dengan `Connect` nama perintah.

Mengkonversi Apache Oozie ke dengan AWS Step FunctionsAWS Schema Conversion Tool

Untuk mengonversi alur kerja Apache Oozie, pastikan Anda menggunakan AWS SCT versi 1.0.671 atau yang lebih tinggi. Juga, membiasakan diri dengan antarmuka baris perintah (CLI) dari AWS SCT Untuk informasi selengkapnya, lihat [AWS SCT Referensi CLI](#).

Topik

- [Ikhtisar konversi](#)
- [Langkah 1: Hubungkan ke sumber dan layanan target Anda](#)
- [Langkah 2: Mengatur aturan pemetaan](#)
- [Langkah 3: Konfigurasi parameter](#)
- [Langkah 4: Buat laporan penilaian](#)
- [Langkah 5: Konversikan alur kerja Apache Oozie Anda AWS Step FunctionsAWS SCT](#)
- [Menjalankan skrip CLI Anda](#)
- [Apache Oozie node yang AWS SCT dapat mengkonversi ke AWS Step Functions](#)

Ikhtisar konversi

Kode sumber Apache Oozie Anda mencakup node tindakan, node alur kontrol, dan properti pekerjaan. Action node menentukan pekerjaan, yang Anda jalankan dalam alur kerja Apache Oozie Anda. Bila Anda menggunakan Apache Oozie untuk mengatur kluster Apache Hadoop Anda, maka node tindakan menyertakan pekerjaan Hadoop. Kontrol node aliran menyediakan mekanisme untuk mengontrol jalur alur kerja. Node aliran kontrol termasuk node seperti `start`, `end`, `decision`, `fork`, dan `join`.

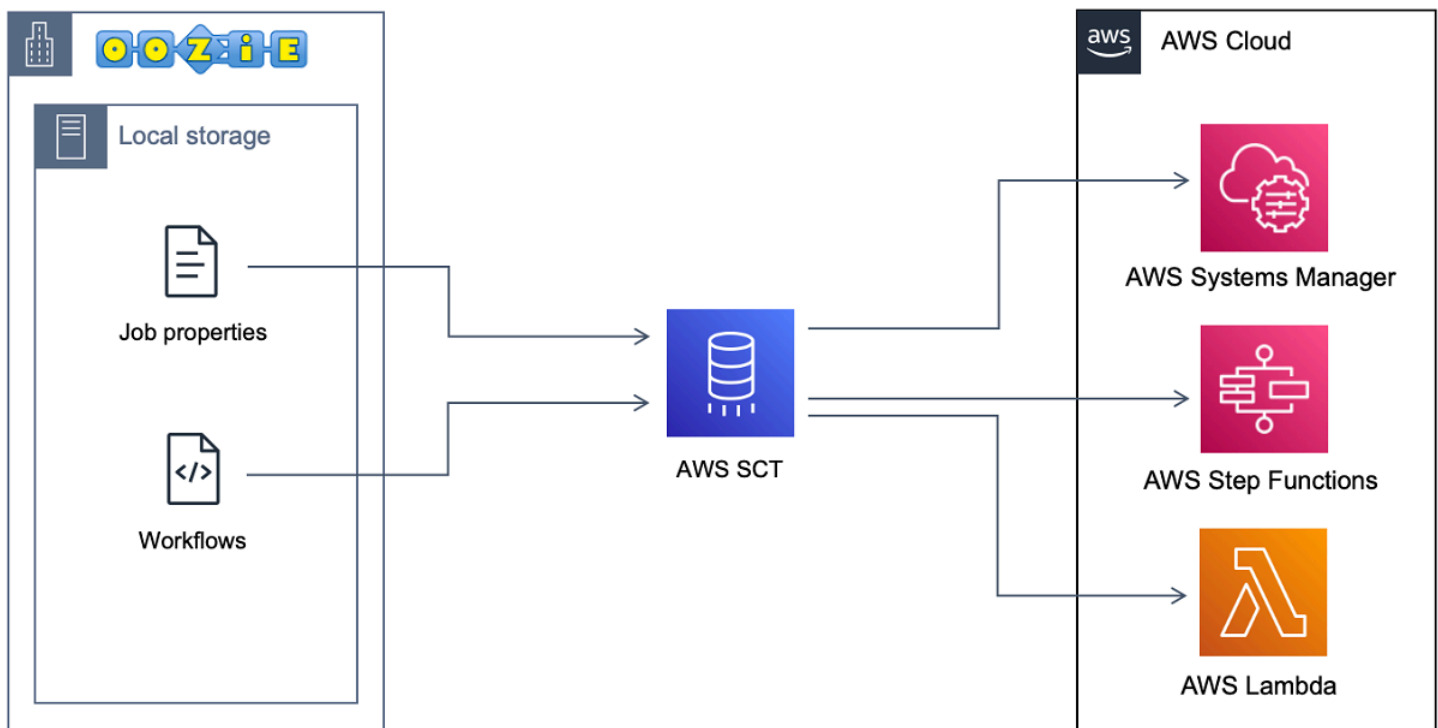
AWS SCT mengkonversi node tindakan sumber Anda dan mengontrol node aliran untuk AWS Step Functions. Di dalamnya AWS Step Functions, Anda menentukan alur kerja Anda di Amazon States Language (ASL). AWS SCT menggunakan ASL untuk menentukan mesin negara Anda, yang merupakan kumpulan negara, yang dapat melakukan pekerjaan, menentukan negara untuk transisi ke berikutnya, berhenti dengan kesalahan, dan sebagainya. Selanjutnya, AWS SCT upload file JSON dengan definisi mesin negara. Kemudian, AWS SCT dapat menggunakan peran AWS Identity and Access Management (IAM) Anda untuk mengonfigurasi mesin negara AWS Step Functions Anda. Untuk informasi lebih lanjut, lihat [Apa yang dimaksud AWS Step Functions?](#) dalam Panduan Developer AWS Step Functions.

Selain itu, AWS SCT buat paket ekstensi dengan AWS Lambda fungsi yang meniru fungsi sumber yang AWS Step Functions tidak mendukung. Untuk informasi selengkapnya, lihat [Menggunakan paket AWS SCT ekstensi](#).

AWS SCT memigrasikan properti pekerjaan sumber Anda ke AWS Systems Manager. Untuk menyimpan nama parameter dan nilai-nilai, AWS SCT menggunakan Parameter Store, kemampuan AWS Systems Manager. Untuk informasi lebih lanjut, lihat [Apa AWS Systems Manager?](#) dalam AWS Systems Manager Panduan Pengguna.

Anda dapat menggunakan AWS SCT untuk memperbarui nilai dan nama parameter Anda secara otomatis. Karena perbedaan arsitektur antara Apache Oozie dan AWS Step Functions, Anda mungkin perlu mengkonfigurasi parameter Anda. AWS SCT dapat menemukan nama parameter tertentu atau nilai dalam file sumber Anda dan menggantinya dengan nilai-nilai baru. Untuk informasi selengkapnya, lihat [Langkah 3: Konfigurasi parameter](#).

Gambar berikut menunjukkan diagram arsitektur konversi Apache Oozie ke AWS Step Functions



Untuk memulai konversi, buat dan jalankan skrip AWS SCT CLI Anda. Script ini mencakup set lengkap perintah untuk menjalankan konversi. Anda dapat mengunduh dan mengedit templat skrip konversi Apache Oozie. Untuk informasi selengkapnya, lihat [Mendapatkan skenario CLI](#).

Pastikan skrip Anda menyertakan langkah-langkah berikut.

Langkah 1: Hubungkan ke sumber dan layanan target Anda

Untuk memulai konversi cluster Apache Oozie Anda, buat proyek baru. AWS SCT Selanjutnya, sambungkan ke sumber dan layanan target Anda. Pastikan Anda membuat dan menyediakan AWS sumber daya target sebelum memulai migrasi. Untuk informasi selengkapnya, lihat [Prasyarat untuk menggunakan Apache Oozie sebagai sumber](#).

Pada langkah ini, Anda menggunakan perintah AWS SCT CLI berikut.

- `CreateProject`— untuk membuat AWS SCT proyek baru.
- `AddSource`— untuk menambahkan file Apache Oozie sumber Anda dalam proyek Anda. AWS SCT
- `ConnectSource`— untuk terhubung ke Apache Oozie sebagai sumber.
- `AddTarget`— untuk menambahkan AWS Step Functions sebagai target migrasi dalam proyek Anda.
- `ConnectTarget`— untuk terhubung ke AWS Step Functions.

Untuk contoh menggunakan perintah AWS SCT CLI ini, lihat [Menggunakan Apache Oozie sebagai sumber](#).

Ketika Anda menjalankan `ConnectSource` atau `ConnectTarget` perintah, AWS SCT mencoba untuk membuat koneksi ke layanan Anda. Jika upaya koneksi gagal, maka AWS SCT berhenti menjalankan perintah dari skrip CLI Anda dan menampilkan pesan kesalahan.

Langkah 2: Mengatur aturan pemetaan

Setelah tersambung ke layanan sumber dan target, siapkan aturan pemetaan. Aturan pemetaan menentukan target migrasi untuk alur kerja dan parameter Apache Oozie sumber Anda. Untuk informasi selengkapnya tentang aturan pemetaan, lihat [Membuat aturan pemetaan di AWS SCT](#).

Untuk menentukan sumber dan objek target untuk konversi, gunakan `AddServerMapping` perintah. Perintah ini menggunakan dua parameter: `sourceTreePath` dan `targetTreePath`. Nilai parameter ini mencakup jalur eksplisit ke sumber dan objek target Anda. Untuk Apache Oozie untuk AWS Step Functions konversi, parameter ini harus dimulai dengan `ETL`

Contoh kode berikut membuat aturan pemetaan untuk `OOZIE` dan `AWS_STEP_FUNCTIONS` objek. Anda menambahkan objek ini ke AWS SCT proyek Anda menggunakan `AddSource` dan `AddTarget` perintah pada langkah sebelumnya.


```
AddServerMapping
  -sourceTreePath: 'ETL.APACHE_OOZIE'
  -targetTreePath: 'ETL.AWS_STEP_FUNCTIONS'
/
```

Untuk informasi lebih lanjut tentang `AddServerMapping` perintah, lihat [Referensi AWS Schema Conversion Tool CLI](#).

Langkah 3: Konfigurasi parameter

Jika alur kerja Apache Oozie sumber Anda menggunakan parameter, Anda mungkin perlu mengubah nilainya setelah konversi ke AWS Step Functions. Juga, Anda mungkin perlu menambahkan parameter baru untuk digunakan dengan AWS Step Functions.

Pada langkah ini, Anda menggunakan `AddParameterMapping` dan `AddTargetParameter` perintah.

Untuk mengganti nilai parameter dalam file sumber Anda, gunakan `AddParameterMapping` perintah. AWS SCT memindai file sumber Anda, menemukan parameter berdasarkan nama atau nilai, dan mengubah nilainya. Anda dapat menjalankan satu perintah untuk memindai semua file sumber Anda. Anda menentukan ruang lingkup file untuk memindai menggunakan salah satu dari tiga parameter pertama dari daftar berikut. Perintah ini menggunakan hingga enam parameter.

- `filterName`- nama filter untuk objek sumber Anda. Anda dapat membuat filter menggunakan `CreateFilter` perintah.
- `treePath`- jalur eksplisit ke objek sumber Anda.
- `namePath`- jalur eksplisit ke objek sumber tertentu.
- `sourceParameterName`- nama parameter sumber Anda.
- `sourceValue`- nilai parameter sumber Anda.
- `targetValue`- nilai parameter target Anda.

Contoh kode berikut menggantikan semua parameter di mana nilai sama `c:\oozie\hive.py` dengan `s3://bucket-oozie/hive.py` nilai.

```
AddParameterMapping
  -treePath: 'ETL.OOZIE.Applications'
  -sourceValue: 'c:\oozie\hive.py'
```

```
-targetValue: 's3://bucket-oozie/hive.py'  
/
```

Contoh kode berikut menggantikan semua parameter di mana nama sama nameNode dengan dengan `hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020` nilai.

```
AddParameterMapping  
-treePath: 'ETL.OOZIE_SOURCE.Applications'  
-sourceParameter: 'nameNode'  
-targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'  
/
```

Contoh kode berikut menggantikan semua parameter di mana nama sama dengan nameNode dan nilai sama `hdfs://ip-55.eu-west-1.compute.internal:8020` dengan dengan nilai dari `targetValue` parameter.

```
AddParameterMapping  
-treePath: 'ETL.OOZIE_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'  
/
```

Untuk menambahkan parameter baru dalam file target Anda selain parameter yang ada dari file sumber Anda, gunakan `AddTargetParameter` perintah. Perintah ini menggunakan set parameter yang sama dengan `AddParameterMapping` perintah.

Contoh kode berikut menambahkan parameter `clusterId` target bukan nameNode parameter.

```
AddTargetParameter  
-treePath: 'ETL.OOZIE_SOURCE.Applications'  
-sourceParameter: 'nameNode'  
-sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'  
-targetParameter: 'clusterId'  
-targetValue: '1234567890abcdef0'  
/
```

Untuk informasi lebih lanjut tentang `AddServerMapping`, `AddParameterMapping`, `AddTargetParameter`, dan `CreateFilter` perintah, lihat [Referensi AWS Schema Conversion Tool CLI](#).

Langkah 4: Buat laporan penilaian

Sebelum memulai konversi, kami sarankan untuk membuat laporan penilaian. Laporan ini merangkum semua tugas migrasi dan merinci item tindakan yang akan muncul selama migrasi. Untuk memastikan migrasi Anda tidak gagal, lihat laporan ini dan alamat item tindakan sebelum migrasi. Untuk informasi selengkapnya, lihat [Laporan penilaian migrasi](#).

Pada langkah ini, Anda menggunakan `CreateReport` perintah. Perintah ini menggunakan dua parameter. Parameter pertama menjelaskan objek sumber yang AWS SCT membuat laporan penilaian. Untuk melakukannya, gunakan salah satu parameter berikut: `filterName`, `treePath`, atau `ataunamPath`. Parameter ini wajib. Selain itu, Anda dapat menambahkan parameter `forceLoad` Boolean opsional. Jika Anda mengatur parameter `inittrue`, maka AWS SCT secara otomatis memuat semua objek anak untuk objek sumber yang Anda tentukan dalam `CreateReport` perintah.

Contoh kode berikut membuat laporan penilaian untuk `Applications` node file Oozie sumber Anda.

```
CreateReport
  -treePath: 'ETL.APACHE_00ZIE.Applications'
/
```

Anda kemudian dapat menyimpan salinan laporan penilaian sebagai file PDF atau nilai dipisahkan koma (CSV). Untuk melakukannya, gunakan `SaveReportCSV` perintah `SaveReportPDF` or.

`SaveReportPDF`Perintah menyimpan salinan laporan penilaian Anda sebagai file PDF. Perintah ini menggunakan empat parameter. `fileParameter`nya wajib, parameter lainnya bersifat opsional.

- `file`- jalur ke file PDF dan namanya.
- `filter`- nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.
- `treePath`— jalur eksplisit ke objek database sumber tempat Anda menyimpan salinan laporan penilaian.
- `namePath`— jalur yang hanya menyertakan nama objek target yang Anda simpan salinan laporan penilaian.

`SaveReportCSV`Perintah menyimpan laporan penilaian Anda dalam file CSV. Perintah ini menggunakan empat parameter. `directoryParameter`nya wajib, parameter lainnya bersifat opsional.

- `directory`- path ke folder tempat AWS SCT menyimpan file CSV.
- `filter`- nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.
- `treePath`— jalur eksplisit ke objek database sumber tempat Anda menyimpan salinan laporan penilaian.
- `namePath`— jalur yang hanya menyertakan nama objek target yang Anda simpan salinan laporan penilaian.

Contoh kode berikut menyimpan salinan laporan penilaian dalam `c:\sct\ar.pdf` file.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

Contoh kode berikut menyimpan salinan laporan penilaian sebagai file CSV di `c:\sct` folder.

```
SaveReportCSV
-file:'c:\sct'
/
```

Untuk informasi lebih lanjut tentang `CreateReport`, `SaveReportPDF` dan `SaveReportCSV` perintah, lihat [Referensi AWS Schema Conversion Tool CLI](#).

Langkah 5: Konversikan alur kerja Apache Oozie Anda AWS Step Functions AWS SCT

Setelah Anda mengkonfigurasi AWS SCT proyek Anda, mengkonversi kode sumber Anda dan menerapkannya ke AWS Cloud.

Pada langkah ini, Anda menggunakan `Convert`, `SaveOnS3`, `ConfigureStateMachine`, dan `ApplyToTarget` perintah.

`MigratePerintah` memigrasikan objek sumber Anda ke cluster target. Perintah ini menggunakan empat parameter. Pastikan bahwa Anda menentukan `filter` atau `treePath` parameter. Parameter lainnya bersifat opsional.

- `filter`- nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.

- `namePath`- jalur eksplisit ke objek sumber tertentu.
- `treePath`— jalur eksplisit ke objek database sumber tempat Anda menyimpan salinan laporan penilaian.
- `forceLoad`- ketika diatur ke `true`, AWS SCT secara otomatis memuat pohon metadata database selama migrasi. Nilai default-nya adalah `false`.

Contoh kode berikut mengkonversi file dari `Applications` folder dalam file Oozie sumber Anda.

```
Convert
  -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

`SaveOnS3` Mengunggah definisi mesin status ke bucket Amazon S3 Anda. Perintah ini menggunakan `treePath` parameter. Untuk menjalankan perintah ini, gunakan folder target dengan definisi mesin negara sebagai nilai parameter ini.

Berikut ini mengunggah `State machine definitions` folder objek `AWS_STEP_FUNCTIONS` target Anda ke bucket Amazon S3. AWS SCT menggunakan bucket Amazon S3 yang Anda simpan di profil AWS layanan di [Prasyarat](#) langkah tersebut.

```
SaveOnS3
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
/
```

`ConfigureStateMachine` Perintah mengkonfigurasi mesin negara. Perintah ini menggunakan hingga enam parameter. Pastikan bahwa Anda menentukan lingkup target menggunakan salah satu dari tiga parameter pertama dari daftar berikut.

- `filterName`- nama filter untuk objek target Anda. Anda dapat membuat filter menggunakan `CreateFilter` perintah.
- `treePath`- jalur eksplisit ke objek target Anda.
- `namePath`- jalur eksplisit ke objek target tertentu.
- `iamRole`— Amazon Resource Name (ARN) dari peran IAM yang menyediakan akses ke mesin langkah Anda. Parameter ini diperlukan.

Contoh kode berikut mengkonfigurasi mesin negara didefinisikan dalam `AWS_STEP_FUNCTIONS` menggunakan peran `role_name` IAM.

```
ConfigureStateMachine
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
  -role: 'arn:aws:iam::555555555555:role/role_name'
/
```

ApplyToTargetPerintah menerapkan kode yang dikonversi ke server target. Untuk menjalankan perintah ini, gunakan salah satu parameter berikut: `filterName`, `treePath`, atau `namePath` untuk menentukan objek target untuk diterapkan.

Contoh kode berikut berlaku mesin `app_wp` negara untuk AWS Step Functions.

```
ApplyToTarget
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machines.app_wp'
/
```

Untuk memastikan bahwa kode yang dikonversi menghasilkan hasil yang sama dengan kode sumber, Anda dapat menggunakan paket AWS SCT ekstensi. Ini adalah satu set AWS Lambda fungsi yang meniru fungsi Apache Oozie Anda yang tidak AWS Step Functions mendukung. Untuk menginstal paket ekstensi ini, Anda dapat menggunakan `CreateLambdaExtPack` perintah.

Perintah ini menggunakan hingga lima parameter. Pastikan yang Anda gunakan **Oozie2SF** untuk `extPackId`. Dalam hal ini, AWS SCT membuat paket ekstensi untuk fungsi sumber Apache Oozie.

- `extPackId`— pengenal unik untuk satu set fungsi Lambda. Parameter ini diperlukan.
- `tempDirectory`— jalan di mana AWS SCT dapat menyimpan file-file sementara. Parameter ini diperlukan.
- `awsProfile`— nama AWS profil Anda.
- `lambdaExecRoles`— daftar Amazon Resource Names (ARN) dari peran eksekusi yang akan digunakan untuk fungsi Lambda.
- `createInvokeRoleFlag`— bendera Boolean yang menunjukkan apakah akan membuat peran eksekusi untuk AWS Step Functions.

Untuk menginstal dan menggunakan paket ekstensi, pastikan Anda memberikan izin yang diperlukan. Untuk informasi selengkapnya, lihat [Izin untuk menggunakan AWS Lambda fungsi dalam paket ekstensi](#).

Untuk informasi lebih lanjut tentang `Convert`, `SaveOnS3`, `ConfigureStateMachine`, `ApplyToTarget`, dan `CreateLambdaExtPack` perintah, lihat [Referensi AWS Schema Conversion Tool CLI](#).

Menjalankan skrip CLI Anda

Setelah Anda selesai mengedit skrip AWS SCT CLI Anda, simpan sebagai file dengan `.scts` ekstensi. Sekarang, Anda dapat menjalankan skrip Anda dari app folder jalur AWS SCT instalasi Anda. Untuk melakukannya, gunakan perintah berikut.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\oozie.scts"
```

Pada contoh sebelumnya, ganti *script_path* dengan *path* ke file Anda dengan skrip CLI. Untuk informasi selengkapnya tentang menjalankan skrip CLI AWS SCT, lihat [Mode skrip](#).

Apache Oozie node yang AWS SCT dapat mengkonversi ke AWS Step Functions

Anda dapat menggunakan AWS SCT untuk mengkonversi Apache Oozie node tindakan dan node aliran kontrol untuk AWS Step Functions

Node tindakan yang didukung meliputi yang berikut ini:

- Aksi sarang
- Tindakan Hive2
- Aksi percikan
- MapReduce Aksi streaming
- Aksi Java
- DistCptindakan
- Tindakan babi
- Tindakan Sqoop
- Aksi FS
- Aksi shell

Node aliran kontrol yang didukung meliputi yang berikut:

- Mulai tindakan
- Akhiri tindakan
- Bunuh tindakan
- Tindakan keputusan
- Aksi garpu
- Bergabunglah dengan tindakan

Menggunakan AWS SCT dengan AWS DMS

Menggunakan AWS SCTagen replikasi dengan AWS DMS

Untuk migrasi database yang sangat besar, Anda dapat menggunakan AWS SCTagen replikasi (`aws-schema-conversion-tool-dms-agent`) untuk menyalin data dari database lokal Anda ke Amazon S3 atau AWS Snowball Perangkat tepi. Agen replikasi bekerja sama dengan AWS DMS dan dapat bekerja di latar belakang sementara AWS SCT ditutup.

Saat bekerja dengan AWS Snowball Edge, AWS SCTagen mereplikasi data ke AWS Snowball perangkat. Perangkat ini kemudian dikirim ke AWS dan data dimuat ke bucket Amazon S3. Selama waktu ini, AWS SCTagen terus berjalan. Agen kemudian mengambil data di Amazon S3 dan menyalin data ke titik akhir target.

Untuk informasi selengkapnya, lihat [Memigrasi data dari gudang data lokal ke Amazon Redshift](#).

Menggunakan AWS SCTagen ekstraksi data dengan AWS DMS

Masuk AWS SCT, Anda dapat menemukan agen ekstraksi data (`aws-schema-conversion-tool-extractor`) yang membantu membuat migrasi dari Apache Cassandra ke Amazon DynamoDB lebih mudah. Cassandra dan DynamoDB adalah database NoSQL, tetapi mereka berbeda dalam arsitektur sistem dan representasi data. Anda dapat menggunakan alur kerja berbasis wizard di AWS SCT untuk mengotomatisasi proses migrasi Cassandra-ke-DynamoDB. AWS SCT terintegrasi dengan AWS Database Migration Service (AWS DMS) untuk melakukan migrasi yang sebenarnya.

Untuk informasi selengkapnya, lihat [Memigrasi data dari gudang data lokal ke Amazon Redshift](#).

Meningkatkan tingkat logging saat menggunakan AWS SCT bersama AWS DMS

Anda dapat meningkatkan tingkat logging saat menggunakan AWS SCT bersama AWS DMS, misalnya jika Anda perlu bekerja dengan AWS Support.

Setelah menginstal AWS SCT dan driver yang dibutuhkan, buka aplikasi dengan memilih AWS SCT ikon. Jika Anda melihat pemberitahuan pembaruan, Anda dapat memilih untuk memperbarui sebelum atau setelah proyek Anda selesai. Jika jendela proyek otomatis terbuka, tutup jendela dan buat proyek secara manual.

Untuk meningkatkan tingkat logging saat menggunakan AWS SCT bersama AWS DMS

1. Pada Pengaturan menu, pilih Pengaturan Global.
2. Di Pengaturan Global jendela, pilih Logging.
3. Untuk Mode debug, pilih Benar.
4. Dari Tingkat pesan bagian, Anda dapat memodifikasi jenis berikut log:
 - Umum
 - Loader
 - Parser
 - Printer
 - Penyelesai
 - Telemetri
 - Konverter

Secara default, semua tingkat pesan diatur ke INFO.

5. Pilih tingkat penebangan untuk setiap jenis tingkat pesan yang ingin Anda ubah:
 - Jejak (logging paling rinci)
 - Debug
 - INFO
 - Peringatan
 - Kesalahan (setidaknya penebangan rinci)
 - Kritis
 - Wajib
6. Memilih Terapkan untuk memodifikasi pengaturan untuk proyek Anda.
7. Memilih OKE untuk menutup Pengaturan Global jendela.

Memigrasi data dari gudang data lokal ke Amazon Redshift

Anda dapat menggunakan AWS SCT agen untuk mengekstrak data dari gudang data lokal dan memigrasikannya ke Amazon Redshift. Agen mengekstrak data Anda dan mengunggah data ke Amazon S3 atau, untuk migrasi skala besar, perangkat Edge. AWS Snowball Anda kemudian dapat menggunakan AWS SCT agen untuk menyalin data ke Amazon Redshift.

Atau, Anda dapat menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan data ke Amazon Redshift. AWS DMS Keuntungannya adalah dukungan replikasi yang sedang berlangsung (perubahan pengambilan data). Namun, untuk meningkatkan kecepatan migrasi data, gunakan beberapa AWS SCT agen secara paralel. Menurut pengujian kami, AWS SCT agen memigrasikan data lebih cepat daripada AWS DMS 15-35 persen. Perbedaan kecepatan adalah karena kompresi data, dukungan migrasi partisi tabel secara paralel, dan pengaturan konfigurasi yang berbeda. Untuk informasi selengkapnya, lihat [Menggunakan database Amazon Redshift sebagai target](#). AWS Database Migration Service

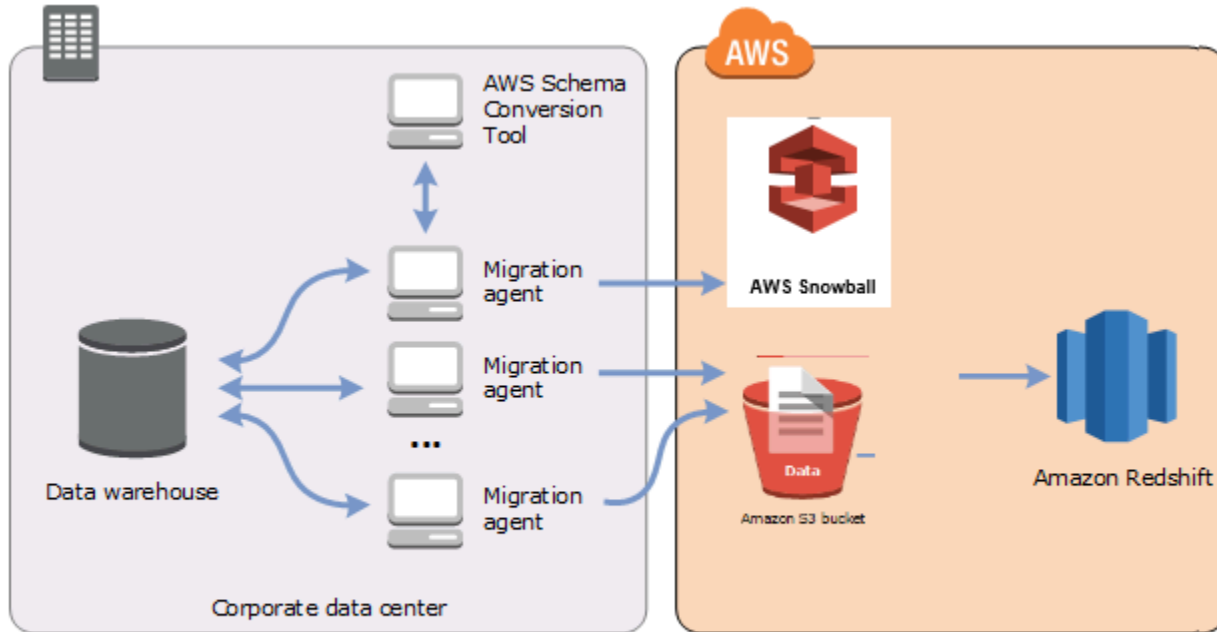
Amazon S3 adalah layanan penyimpanan dan pengambilan. Untuk menyimpan objek di Amazon S3, Anda mengunggah file yang ingin Anda simpan ke bucket Amazon S3. Saat Anda mengunggah file, Anda dapat mengatur izin pada objek dan juga pada metadata apa pun.

Migrasi skala besar

Migrasi data skala besar dapat mencakup banyak terabyte informasi, dan dapat diperlambat oleh kinerja jaringan dan oleh banyaknya data yang harus dipindahkan. AWS Snowball Edge adalah AWS layanan yang dapat Anda gunakan untuk mentransfer data ke cloud dengan faster-than-network kecepatan menggunakan alat yang AWS dimiliki. Perangkat AWS Snowball Edge dapat menampung hingga 100 TB data. Menggunakan enkripsi 256-bit dan Trusted Platform Module (TPM) standar industri untuk memastikan keamanan dan penuh untuk data Anda. chain-of-custody AWS SCT bekerja dengan perangkat AWS Snowball Edge.

Saat Anda menggunakan AWS SCT dan perangkat AWS Snowball Edge, Anda memigrasikan data dalam dua tahap. Pertama, Anda gunakan AWS SCT untuk memproses data secara lokal dan kemudian memindahkan data itu ke perangkat AWS Snowball Edge. Anda kemudian mengirim perangkat untuk AWS menggunakan proses AWS Snowball Edge, dan kemudian AWS secara otomatis memuat data ke dalam bucket Amazon S3. Selanjutnya, saat data tersedia di Amazon S3, Anda gunakan AWS SCT untuk memigrasikan data ke Amazon Redshift. Agen ekstraksi data dapat bekerja di latar belakang saat AWS SCT ditutup.

Diagram berikut menunjukkan skenario yang didukung.



Agen ekstraksi data saat ini didukung untuk gudang data sumber berikut:

- Analisis Sinaps Azure
- BigQuery
- Database Greenplum (versi 4.3)
- Microsoft SQL Server (versi 2008 dan lebih tinggi)
- Netezza (versi 7.0.3 dan lebih tinggi)
- Oracle (versi 10 dan lebih tinggi)
- Kepingan salju (versi 3)
- Teradata (versi 13 dan lebih tinggi)
- Vertica (versi 7.2.2 dan lebih tinggi)

Anda dapat terhubung ke titik akhir FIPS untuk Amazon Redshift jika Anda perlu mematuhi persyaratan keamanan Federal Information Processing Standard (FIPS). Titik akhir FIPS tersedia di Wilayah berikut: AWS

- Wilayah AS Timur (Virginia N.) (redshift-fips.us-east-1.amazonaws.com)

- Wilayah Timur AS (Ohio) (redshift-fips.us-east-2.amazonaws.com)
- Wilayah AS Barat (California N.) (redshift-fips.us-west-1.amazonaws.com)
- Wilayah Barat AS (Oregon) (redshift-fips.us-west-2.amazonaws.com)

Gunakan informasi dalam topik berikut untuk mempelajari cara bekerja dengan agen ekstraksi data.

Topik

- [Prasyarat untuk menggunakan agen ekstraksi data](#)
- [Instalasi agen ekstraksi](#)
- [Mengkonfigurasi agen ekstraksi](#)
- [Mendaftarkan agen ekstraksi dengan AWS Schema Conversion Tool](#)
- [Menyembunyikan dan memulihkan informasi untuk agen AWS SCT](#)
- [Membuat aturan migrasi data di AWS SCT](#)
- [Mengubah pengaturan ekstraktor dan salin dari pengaturan proyek](#)
- [Menyortir data sebelum bermigrasi menggunakan AWS SCT](#)
- [Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data](#)
- [Mengekspor dan mengimpor tugas ekstraksi AWS SCT data](#)
- [Ekstraksi data menggunakan perangkat AWS Snowball Edge](#)
- [Output tugas ekstraksi data](#)
- [Menggunakan partisi virtual dengan AWS Schema Conversion Tool](#)
- [Menggunakan partisi asli](#)
- [Migrasi LOB ke Amazon Redshift](#)
- [Praktik terbaik dan pemecahan masalah untuk agen ekstraksi data](#)

Prasyarat untuk menggunakan agen ekstraksi data

Sebelum Anda bekerja dengan agen ekstraksi data, tambahkan izin yang diperlukan untuk Amazon Redshift sebagai target ke pengguna Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Izin untuk Amazon Redshift sebagai target](#).

Kemudian, simpan informasi bucket Amazon S3 Anda dan siapkan kepercayaan dan penyimpanan kunci Secure Sockets Layer (SSL) Anda.

Pengaturan Amazon S3

Setelah agen Anda mengekstrak data Anda, mereka mengunggahnya ke bucket Amazon S3 Anda. Sebelum melanjutkan, Anda harus memberikan kredensial untuk terhubung ke AWS akun Anda dan bucket Amazon S3 Anda. Anda menyimpan kredensial dan informasi bucket Anda di profil di pengaturan aplikasi global, lalu mengaitkan profil dengan proyek Anda AWS SCT . Jika perlu, pilih Pengaturan global untuk membuat profil baru. Untuk informasi selengkapnya, lihat [Menyimpan profil AWS layanan di AWS SCT](#).

Untuk memigrasikan data ke database Amazon Redshift target, AWS SCT agen ekstraksi data memerlukan izin untuk mengakses bucket Amazon S3 atas nama Anda. Untuk memberikan izin ini, buat pengguna AWS Identity and Access Management (IAM) dengan kebijakan berikut.

```
{
  "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"
    }
]
```

Pada contoh sebelumnya, ganti *bucket_name* dengan nama bucket Amazon S3 Anda. Kemudian, ganti *111122223333:user/DataExtractionAgentName* dengan nama pengguna IAM Anda.

Dengan asumsi peran IAM

Untuk keamanan tambahan, Anda dapat menggunakan peran AWS Identity and Access Management (IAM) untuk mengakses bucket Amazon S3. Untuk melakukannya, buat pengguna IAM untuk agen ekstraksi data Anda tanpa izin apa pun. Kemudian, buat peran IAM yang memungkinkan akses Amazon S3, dan tentukan daftar layanan dan pengguna yang dapat mengambil peran ini. Untuk informasi selengkapnya, lihat [peran IAM](#) dalam Panduan Pengguna IAM.

Untuk mengonfigurasi peran IAM untuk mengakses bucket Amazon S3

1. Buat pengguna IAM baru. Untuk kredensi pengguna, pilih Jenis akses terprogram.
2. Konfigurasi lingkungan host sehingga agen ekstraksi data Anda dapat mengambil peran yang AWS SCT menyediakan. Pastikan bahwa pengguna yang Anda konfigurasi pada langkah sebelumnya memungkinkan agen ekstraksi data untuk menggunakan rantai penyedia kredensi. Untuk informasi selengkapnya, lihat [Menggunakan kredensial](#) di Panduan AWS SDK for Java Pengembang.
3. Buat peran IAM baru yang memiliki akses ke bucket Amazon S3 Anda.
4. Ubah bagian kepercayaan dari peran ini untuk mempercayai pengguna yang Anda buat sebelumnya untuk mengambil peran tersebut. Dalam contoh berikut, ganti *111122223333:user/DataExtractionAgentName* dengan nama pengguna Anda.

```
{
```

```
"Effect": "Allow",
"Principal": {
  "AWS": "arn:aws:iam::111122223333:user/DataExtractionAgentName"
},
"Action": "sts:AssumeRole"
}
```

- Ubah bagian kepercayaan dari peran ini menjadi kepercayaan `redshift.amazonaws.com` untuk mengambil peran.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "redshift.amazonaws.com"
    ]
  },
  "Action": "sts:AssumeRole"
}
```

- Lampirkan peran ini ke cluster Amazon Redshift Anda.

Sekarang, Anda dapat menjalankan agen ekstraksi data Anda di AWS SCT.

Bila Anda menggunakan asumsi peran IAM, migrasi data berfungsi dengan cara berikut. Agen ekstraksi data memulai dan mendapatkan kredensial pengguna menggunakan rantai penyedia kredensi. Selanjutnya, Anda membuat tugas migrasi data di AWS SCT, lalu tentukan peran IAM untuk diasumsikan oleh agen ekstraksi data, dan memulai tugas. AWS Security Token Service (AWS STS) menghasilkan kredensial sementara untuk mengakses Amazon S3. Agen ekstraksi data menggunakan kredensial ini untuk mengunggah data ke Amazon S3.

Kemudian, AWS SCT berikan Amazon Redshift dengan peran IAM. Pada gilirannya, Amazon Redshift mendapatkan kredensial sementara baru dari untuk mengakses AWS STS Amazon S3. Amazon Redshift menggunakan kredensial ini untuk menyalin data dari Amazon S3 ke tabel Amazon Redshift Anda.

Pengaturan keamanan

Agen ekstraksi AWS Schema Conversion Tool dan ekstraksi dapat berkomunikasi melalui Secure Sockets Layer (SSL). Untuk mengaktifkan SSL, siapkan toko kepercayaan dan toko kunci.

Untuk mengatur komunikasi yang aman dengan agen ekstraksi Anda

1. Mulai AWS Schema Conversion Tool.
2. Buka menu Pengaturan, lalu pilih Pengaturan global. Kotak dialog Pengaturan global muncul.
3. Pilih Keamanan.
4. Pilih Hasilkan kepercayaan dan toko kunci, atau pilih Pilih toko kepercayaan yang ada.

Jika Anda memilih Hasilkan kepercayaan dan penyimpanan kunci, Anda kemudian menentukan nama dan kata sandi untuk kepercayaan dan penyimpanan kunci, dan jalur ke lokasi untuk file yang dihasilkan. Anda menggunakan file-file ini di langkah selanjutnya.

Jika Anda memilih Pilih toko kepercayaan yang ada, Anda kemudian menentukan kata sandi dan nama file untuk kepercayaan dan penyimpanan kunci. Anda menggunakan file-file ini di langkah selanjutnya.

5. Setelah Anda menentukan toko kepercayaan dan toko kunci, pilih OK untuk menutup kotak dialog Pengaturan global.

Mengkonfigurasi lingkungan untuk agen ekstraksi data

Anda dapat menginstal beberapa agen ekstraksi data pada satu host. Namun, kami menyarankan Anda menjalankan satu agen ekstraksi data pada satu host.

Untuk menjalankan agen ekstraksi data Anda, pastikan Anda menggunakan host dengan setidaknya empat vCPU dan memori 32 GB. Juga, atur memori minimum yang tersedia AWS SCT untuk setidaknya empat GB. Untuk informasi selengkapnya, lihat [Mengkonfigurasi memori tambahan](#).

Konfigurasi optimal dan jumlah host agen tergantung pada situasi spesifik setiap pelanggan. Pastikan Anda mempertimbangkan faktor-faktor seperti jumlah data yang akan dimigrasi, bandwidth jaringan, waktu untuk mengekstrak data, dan sebagainya. Anda dapat melakukan bukti konsep (PoC) terlebih dahulu, dan kemudian mengkonfigurasi agen ekstraksi data dan host Anda sesuai dengan hasil PoC ini.

Instalasi agen ekstraksi

Kami menyarankan Anda menginstal beberapa agen ekstraksi pada komputer individu, terpisah dari komputer yang menjalankan AWS Schema Conversion Tool.

Agen ekstraksi saat ini didukung pada sistem operasi berikut:

- Microsoft Windows
- Perusahaan Topi Merah Linux (RHEL) 6.0
- Ubuntu Linux (versi 14.04 dan lebih tinggi)

Gunakan prosedur berikut untuk menginstal agen ekstraksi. Ulangi prosedur ini untuk setiap komputer tempat Anda ingin menginstal agen ekstraksi.

Untuk memasang agen ekstraksi

1. Jika Anda belum mengunduh file AWS SCT penginstal, ikuti instruksi di [Menginstal, memverifikasi, dan memperbarui AWS SCT](#) untuk mengunduhnya. File.zip yang berisi file AWS SCT installer juga berisi file installer agen ekstraksi.
2. Unduh dan instal versi terbaru Amazon Corretto 11. Untuk informasi selengkapnya, lihat [Unduhan untuk Amazon Corretto 11 di Panduan Pengguna Amazon Corretto 11](#).
3. Temukan file penginstal untuk agen ekstraksi Anda di subfolder bernama agen. Untuk setiap sistem operasi komputer, file yang benar untuk menginstal agen ekstraksi ditunjukkan sebagai berikut.

Sistem operasi	Nama file
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

4. Instal agen ekstraksi pada komputer terpisah dengan menyalin file penginstal ke komputer baru.
5. Jalankan file installer. Gunakan instruksi untuk sistem operasi Anda, yang ditunjukkan berikut.

Sistem operasi	Petunjuk pemasangan
Microsoft Windows	Klik dua kali file untuk menjalankan penginstal.

Sistem operasi	Petunjuk pemasangan
RHEL	<p>Jalankan perintah berikut di folder yang Anda unduh atau pindahkan file.</p> <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>
Ubuntu Linux	<p>Jalankan perintah berikut di folder yang Anda unduh atau pindahkan file.</p> <pre>sudo dpkg -i aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .deb sudo ./sct-extractor-setup.sh --config</pre>

- Pilih Berikutnya, terima perjanjian lisensi, dan pilih Berikutnya.
- Masukkan jalur untuk menginstal agen ekstraksi AWS SCT data, dan pilih Berikutnya.
- Pilih Instal untuk menginstal agen ekstraksi data Anda.

AWS SCT menginstal agen ekstraksi data Anda. Untuk menyelesaikan instalasi, konfigurasi agen ekstraksi data Anda. AWS SCT secara otomatis meluncurkan program pengaturan konfigurasi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi agen ekstraksi](#).

- Pilih Selesai untuk menutup wizard instalasi setelah Anda mengonfigurasi agen ekstraksi data Anda.

Mengkonfigurasi agen ekstraksi

Gunakan prosedur berikut untuk mengkonfigurasi agen ekstraksi. Ulangi prosedur ini di setiap komputer yang memiliki agen ekstraksi terpasang.

Untuk mengkonfigurasi agen ekstraksi Anda

- Luncurkan program pengaturan konfigurasi:
 - Di Windows, AWS SCT meluncurkan program pengaturan konfigurasi secara otomatis selama instalasi agen ekstraksi data.

Sesuai kebutuhan, Anda dapat meluncurkan program pengaturan secara manual. Untuk melakukannya, jalankan `ConfigAgent.bat` file di Windows. Anda dapat menemukan file ini di folder tempat Anda menginstal agen.

- Di RHEL dan Ubuntu, jalankan `sct-extractor-setup.sh` file dari lokasi tempat Anda menginstal agen.

Program pengaturan meminta Anda untuk informasi. Untuk setiap prompt, nilai default muncul.

2. Terima nilai default pada setiap prompt, atau masukkan nilai baru.

Tentukan informasi berikut:

- Untuk port Listening, masukkan nomor port yang didengarkan agen.
- Untuk Tambahkan vendor sumber, masukkan ya, lalu masukkan platform gudang data sumber Anda.
- Untuk driver JDBC, masukkan lokasi tempat Anda menginstal driver JDBC.
- Untuk folder Kerja, masukkan jalur tempat agen ekstraksi AWS SCT data akan menyimpan data yang diekstraksi. Folder kerja dapat berada di komputer yang berbeda dari agen, dan satu folder kerja dapat dibagikan oleh beberapa agen di komputer yang berbeda.
- Untuk Aktifkan komunikasi SSL, masukkan ya.
- Untuk Key Store, masukkan lokasi file penyimpanan kunci.
- Untuk kata sandi Key Store, masukkan kata sandi untuk toko kunci.
- Untuk Aktifkan otentikasi SSL klien, masukkan ya.
- Untuk Trust store, masukkan lokasi file trust store.
- Untuk kata sandi toko Trust, masukkan kata sandi untuk toko kepercayaan.

Program pengaturan memperbarui file pengaturan untuk agen ekstraksi. File pengaturan diberi `namasettings.properties`, dan terletak di mana Anda menginstal agen ekstraksi.

Berikut ini adalah contoh file pengaturan.

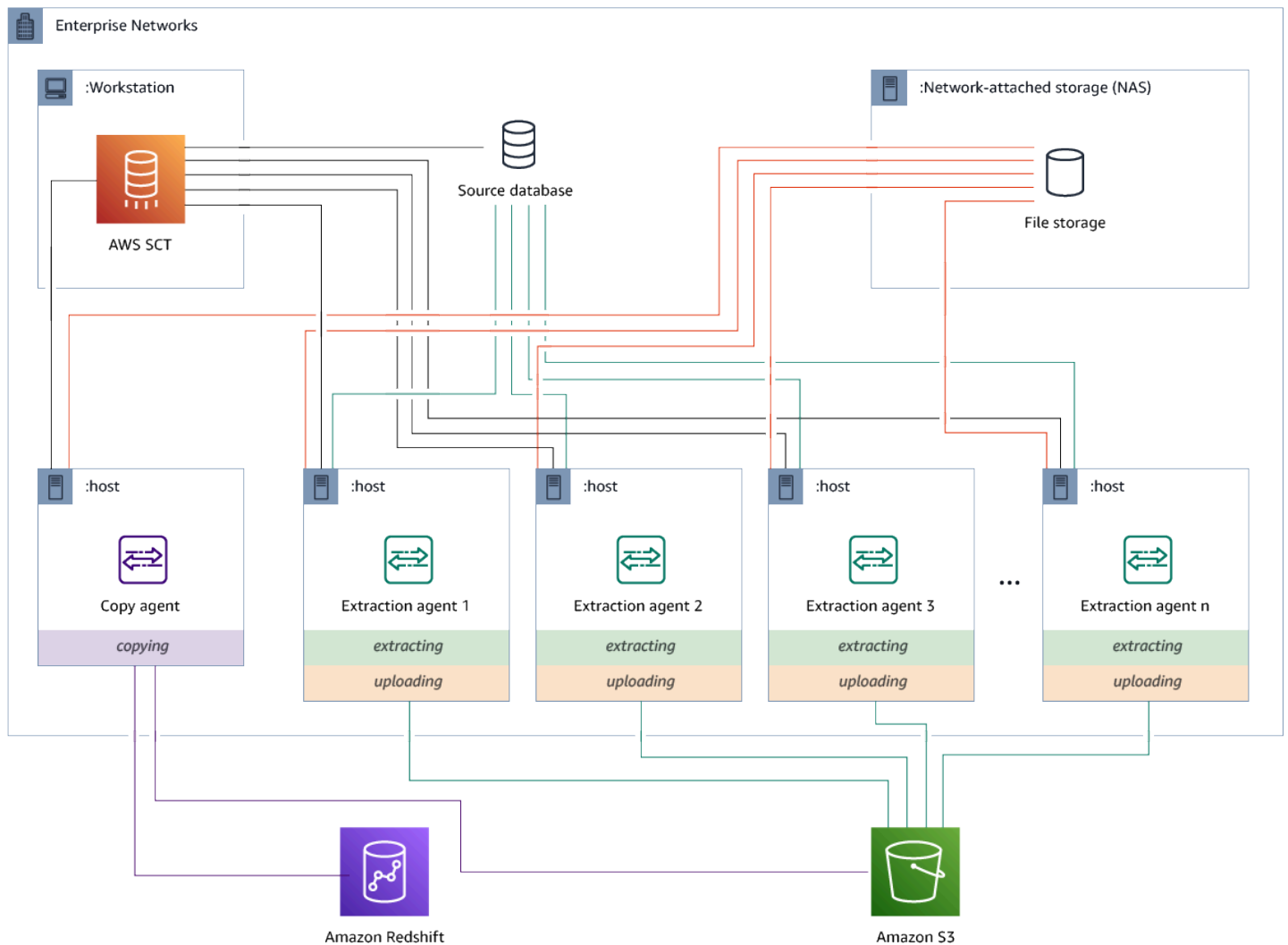
```
$ cat settings.properties
#extractor.start.fetch.size=20000
#extractor.out.file.size=10485760
#extractor.source.connection.pool.size=20
#extractor.source.connection.pool.min.evictable.idle.time.millis=30000
```

```
#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
```

Untuk mengubah pengaturan konfigurasi, Anda dapat mengedit `settings.properties` file menggunakan editor teks atau menjalankan konfigurasi agen lagi.

Memasang dan mengonfigurasi agen ekstraksi dengan agen penyalinan khusus

Anda dapat menginstal agen ekstraksi dalam konfigurasi yang memiliki penyimpanan bersama dan agen penyalinan khusus. Diagram berikut menggambarkan skenario ini.



Konfigurasi itu dapat berguna ketika server database sumber mendukung hingga 120 koneksi, dan jaringan Anda memiliki penyimpanan yang cukup. Gunakan prosedur berikut untuk mengonfigurasi agen ekstraksi yang memiliki agen penyalinan khusus.

Untuk menginstal dan mengonfigurasi agen ekstraksi dan agen penyalinan khusus

1. Pastikan bahwa direktori kerja semua agen ekstraksi menggunakan folder yang sama pada penyimpanan bersama.
2. Instal agen ekstraktor dengan mengikuti langkah-langkahnya. [Instalasi agen ekstraksi](#)
3. Konfigurasi agen ekstraksi dengan mengikuti langkah-langkahnya [Mengkonfigurasi agen ekstraksi](#), tetapi tentukan hanya driver JDBC sumber.
4. Konfigurasi agen penyalinan khusus dengan mengikuti langkah-langkahnya [Mengkonfigurasi agen ekstraksi](#), tetapi tentukan hanya driver Amazon Redshift JDBC.

Memulai agen ekstraksi

Gunakan prosedur berikut untuk memulai agen ekstraksi. Ulangi prosedur ini di setiap komputer yang memiliki agen ekstraksi terpasang.

Agan ekstraksi bertindak sebagai pendengar. Ketika Anda memulai agen dengan prosedur ini, agen mulai mendengarkan instruksi. Anda mengirim instruksi agen untuk mengekstrak data dari gudang data Anda di bagian selanjutnya.

Untuk memulai agen ekstraksi Anda

- Pada komputer yang memiliki agen ekstraksi diinstal, jalankan perintah yang tercantum berikut untuk sistem operasi Anda.

Sistem operasi	Mulai perintah
Microsoft Windows	Klik dua kali file <code>StartAgent.bat</code> batch.
RHEL	Jalankan perintah berikut di jalur ke folder yang Anda instal agen: <code>sudo initctl <i>start</i> sct-extractor</code>
Ubuntu Linux	Jalankan perintah berikut di jalur ke folder yang Anda instal agen. Gunakan perintah yang sesuai untuk versi Ubuntu Anda. Ubuntu 14.04: <code>sudo initctl <i>start</i> sct-extractor</code> Ubuntu 15.04 dan lebih tinggi: <code>sudo systemctl <i>start</i> sct-extractor</code>

Untuk memeriksa status agen, jalankan perintah yang sama tetapi ganti `start` dengan `status`.

Untuk menghentikan agen, jalankan perintah yang sama tetapi ganti `start` dengan `stop`.

Mendaftarkan agen ekstraksi dengan AWS Schema Conversion Tool

Anda mengelola agen ekstraksi Anda dengan menggunakan AWS SCT. Agen ekstraksi bertindak sebagai pendengar. Ketika mereka menerima instruksi dari AWS SCT, mereka mengekstrak data dari gudang data Anda.

Gunakan prosedur berikut untuk mendaftarkan agen ekstraksi dengan AWS SCT proyek Anda.

Untuk mendaftarkan agen ekstraksi

1. Mulai AWS Schema Conversion Tool, dan buka proyek.
2. Buka menu Tampilan, lalu pilih tampilan Migrasi Data (lainnya). Tab Agen muncul. Jika Anda memiliki agen terdaftar sebelumnya AWS SCT, tampilkan mereka dalam kotak di bagian atas tab.
3. Pilih Pendaftaran.

Setelah Anda mendaftarkan agen dengan AWS SCT proyek, Anda tidak dapat mendaftarkan agen yang sama dengan proyek yang berbeda. Jika Anda tidak lagi menggunakan agen dalam sebuah AWS SCT proyek, Anda dapat membatalkan pendaftarannya. Anda kemudian dapat mendaftarkannya dengan proyek yang berbeda.

4. Pilih agen data Redshift, lalu pilih OK.
5. Masukkan informasi Anda pada tab Koneksi pada kotak dialog:
 - a. Untuk Deskripsi, masukkan deskripsi agen.
 - b. Untuk Nama Host, masukkan nama host atau alamat IP komputer agen.
 - c. Untuk Port, masukkan nomor port yang didengarkan agen.
 - d. Pilih Daftar untuk mendaftarkan agen dengan AWS SCT proyek Anda.
6. Ulangi langkah sebelumnya untuk mendaftarkan beberapa agen dengan AWS SCT proyek Anda.

Menyembunyikan dan memulihkan informasi untuk agen AWS SCT

AWS SCT Agen mengenkripsi sejumlah besar informasi, misalnya kata sandi ke toko kepercayaan kunci pengguna, akun basis data, informasi AWS akun, dan item serupa. Ia melakukannya dengan

menggunakan file khusus yang disebut `seed.dat`. Secara default, agen membuat file ini di folder kerja pengguna yang pertama kali mengkonfigurasi agen.

Karena pengguna yang berbeda dapat mengkonfigurasi dan menjalankan agen, jalur ke `seed.dat` disimpan dalam `{extractor.private.folder}` parameter `settings.properties` file. Ketika agen mulai, ia dapat menggunakan jalur ini untuk menemukan `seed.dat` file untuk mengakses informasi penyimpanan `key-trust` untuk database tempat ia bertindak.

Anda mungkin perlu memulihkan kata sandi yang disimpan agen dalam kasus ini:

- Jika pengguna kehilangan `seed.dat` file dan lokasi serta port AWS SCT agen tidak berubah.
- Jika pengguna kehilangan `seed.dat` file dan lokasi dan port AWS SCT agen telah berubah. Dalam hal ini, perubahan biasanya terjadi karena agen dimigrasikan ke host atau port lain dan informasi dalam `seed.dat` file tidak lagi valid.

Dalam kasus ini, jika agen dimulai tanpa SSL, itu dimulai dan kemudian mengakses penyimpanan agen yang dibuat sebelumnya. Kemudian pergi ke keadaan Menunggu pemulihan.

Namun, dalam kasus ini, jika agen dimulai dengan SSL, Anda tidak dapat memulai ulang. Ini karena agen tidak dapat mendekripsi kata sandi ke sertifikat yang disimpan dalam file `settings.properties`. Dalam jenis startup ini, agen gagal memulai. Kesalahan yang mirip dengan berikut ini ditulis dalam log: "Agen tidak dapat memulai dengan mode SSL diaktifkan. Harap konfigurasi ulang agen. Alasan: Kata sandi untuk keystore tidak benar.

Untuk memperbaikinya, buat agen baru dan konfigurasi agen untuk menggunakan kata sandi yang ada untuk mengakses sertifikat SSL. Untuk melakukannya, gunakan prosedur berikut.

Setelah Anda melakukan prosedur ini, agen harus menjalankan dan pergi ke status Menunggu pemulihan. AWS SCT secara otomatis mengirimkan kata sandi yang diperlukan ke agen dalam keadaan Menunggu pemulihan. Ketika agen memiliki kata sandi, ia memulai ulang tugas apa pun. Tidak ada tindakan pengguna lebih lanjut yang diperlukan di AWS SCT samping.

Untuk mengkonfigurasi ulang agen dan mengembalikan kata sandi untuk mengakses sertifikat SSL

1. Instal AWS SCT agen baru dan jalankan konfigurasi.
2. Ubah `agent.name` properti dalam `instance.properties` file menjadi nama agen tempat penyimpanan dibuat, agar agen baru bekerja dengan penyimpanan agen yang ada.

`instance.properties`File disimpan dalam folder pribadi agen, yang dinamai menggunakan konvensi berikut:`{output.folder}\dmt\{hostName}_{portNumber}\`.

3. Ubah nama `{output.folder}` ke folder keluaran agen sebelumnya.

Pada titik ini, AWS SCT masih mencoba untuk mengakses extractor lama di host lama dan port. Akibatnya, ekstraktor yang tidak dapat diakses mendapatkan status GAGAL. Anda kemudian dapat mengubah host dan port.

4. Ubah host, port, atau keduanya untuk agen lama dengan menggunakan perintah Modify untuk mengarahkan aliran permintaan ke agen baru.

Ketika AWS SCT dapat melakukan ping ke agen baru, AWS SCT menerima status Menunggu pemulihan dari agen. AWS SCT kemudian secara otomatis memulihkan kata sandi untuk agen.

Setiap agen yang bekerja dengan penyimpanan agen memperbarui file khusus yang disebut `storage.lck` terletak di `{output.folder}\{agentName}\storage\`. File ini berisi ID jaringan agen dan waktu penyimpanan terkunci. Ketika agen bekerja dengan penyimpanan agen, ia memperbarui `storage.lck` file dan memperpanjang sewa penyimpanan dengan 10 menit setiap 5 menit. Tidak ada contoh lain yang dapat bekerja dengan penyimpanan agen ini sebelum masa sewa berakhir.

Membuat aturan migrasi data di AWS SCT

Sebelum Anda mengekstrak data Anda dengan AWS Schema Conversion Tool, Anda dapat mengatur filter yang mengurangi jumlah data yang Anda ekstrak. Anda dapat membuat aturan migrasi data dengan menggunakan WHERE klausa untuk mengurangi data yang Anda ekstrak. Misalnya, Anda dapat menulis WHERE klausa yang memilih data dari satu tabel.

Anda dapat membuat aturan migrasi data dan menyimpan filter sebagai bagian dari proyek Anda. Dengan proyek Anda terbuka, gunakan prosedur berikut untuk membuat aturan migrasi data.

Untuk membuat aturan migrasi data

1. Buka menu Tampilan, lalu pilih tampilan Migrasi Data (lainnya).
2. Pilih Aturan migrasi data, lalu pilih Tambahkan aturan baru.
3. Konfigurasi aturan migrasi data Anda:
 - a. Untuk Nama, masukkan nama untuk aturan migrasi data Anda.

- b. Untuk Di mana nama skema seperti, masukkan filter untuk diterapkan ke skema. Dalam filter ini, WHERE klausa dievaluasi dengan menggunakan klausa. LIKE Untuk memilih satu skema, masukkan nama skema yang tepat. Untuk memilih beberapa skema, gunakan karakter “%” sebagai wildcard untuk mencocokkan sejumlah karakter dalam nama skema.
 - c. Untuk nama tabel seperti, masukkan filter untuk diterapkan ke tabel. Dalam filter ini, WHERE klausa dievaluasi dengan menggunakan klausa. LIKE Untuk memilih satu tabel, masukkan nama yang tepat. Untuk memilih beberapa tabel, gunakan karakter “%” sebagai wildcard untuk mencocokkan sejumlah karakter dalam nama tabel.
 - d. Untuk klausa Where, masukkan WHERE klausa untuk memfilter data.
4. Setelah mengonfigurasi filter, pilih Simpan untuk menyimpan filter, atau Batalkan untuk membatalkan perubahan.
 5. Setelah Anda selesai menambahkan, mengedit, dan menghapus filter, pilih Simpan semua untuk menyimpan semua perubahan Anda.

Untuk mematikan filter tanpa menghapusnya, gunakan ikon sakelar. Untuk menduplikasi filter yang ada, gunakan ikon salin. Untuk menghapus filter yang ada, gunakan ikon hapus. Untuk menyimpan perubahan apa pun yang Anda buat pada filter, pilih Simpan semua.

Mengubah pengaturan ekstraktor dan salin dari pengaturan proyek

Dari jendela Pengaturan proyek AWS SCT, Anda dapat memilih pengaturan untuk agen ekstraksi data dan perintah Amazon RedshiftCOPY.

Untuk memilih pengaturan ini, pilih Pengaturan, Pengaturan proyek, lalu pilih Migrasi data. Di sini, Anda dapat mengedit pengaturan Ekstraksi, pengaturan Amazon S3, dan pengaturan Salin.

Gunakan petunjuk dalam tabel berikut untuk memberikan informasi untuk pengaturan Ekstraksi.

Untuk parameter ini	Lakukan hal berikut
Format kompresi	Tentukan format kompresi file input. Pilih salah satu opsi berikut: GZIP, BZIP2, ZSTD, atau Tidak ada kompresi.
Karakter pembatas	Tentukan karakter ASCII yang memisahkan bidang dalam file input. Karakter nonprinting tidak didukung.

Untuk parameter ini	Lakukan hal berikut
Nilai NULL sebagai string	Aktifkan opsi ini jika data Anda menyertakan terminator null. Jika opsi ini dimatikan, COPY perintah Amazon Redshift memperlakukan null sebagai akhir dari catatan dan mengakhiri proses pemuatan.
Strategi penyortiran	Gunakan penyortiran untuk memulai kembali ekstraksi dari titik kegagalan. Pilih salah satu strategi penyortiran berikut: Gunakan penyortiran setelah kegagalan pertama (disarankan), Gunakan penyortiran jika memungkinkan, atau Jangan pernah menggunakan penyortiran. Untuk informasi selengkapnya, lihat the section called “Menyortir data” .
Skema suhu sumber	Masukkan nama skema dalam database sumber, di mana agen ekstraksi dapat membuat objek sementara.
Ukuran file keluar (dalam MB)	Masukkan ukuran, dalam MB, file yang diunggah ke Amazon S3.
Ukuran file Snowball out (dalam MB)	Masukkan ukuran, dalam MB, file yang diunggah ke AWS Snowball. File bisa berukuran 1—1.000 MB.
Gunakan partisi otomatis. Untuk Greenplum dan Netezza, masukkan ukuran minimal tabel yang didukung (dalam megabyte)	Aktifkan opsi ini untuk menggunakan partisi tabel, lalu masukkan ukuran tabel untuk dipartisi untuk database sumber Greenplum dan Netezza. Untuk migrasi Oracle ke Amazon Redshift, Anda dapat membiarkan bidang ini kosong AWS SCT karena membuat subtask untuk semua tabel yang dipartisi.
Ekstrak LOB	Aktifkan opsi ini untuk mengekstrak objek besar (LOB) dari database sumber Anda. LOB termasuk BLOB, CLOB, NClob, file XML, dan sebagainya. Untuk setiap LOB, agen AWS SCT ekstraksi membuat file data.
Folder LOB ember Amazon S3	Masukkan lokasi agen AWS SCT ekstraksi untuk menyimpan LOB.
Terapkan RTRIM ke kolom string	Aktifkan opsi ini untuk memangkas sekumpulan karakter tertentu dari akhir string yang diekstraksi.

Untuk parameter ini	Lakukan hal berikut
Simpan file secara lokal setelah diunggah ke Amazon S3	Aktifkan opsi ini untuk menyimpan file di komputer lokal Anda setelah agen ekstraksi data mengunggahnya ke Amazon S3.

Gunakan petunjuk dalam tabel berikut untuk memberikan informasi setelah Amazon S3.

Untuk parameter ini	Lakukan hal berikut
Gunakan proxy	Aktifkan opsi ini untuk menggunakan server proxy untuk mengunggah data ke Amazon S3. Kemudian pilih protokol transfer data, masukkan nama host, port, nama pengguna, dan kata sandi.
Jenis titik akhir	Pilih FIPS untuk menggunakan titik akhir Federal Information Processing Standard (FIPS). Pilih VPCE untuk menggunakan titik akhir virtual private cloud (VPC). Kemudian untuk titik akhir VPC, masukkan Domain Name System (DNS) dari titik akhir VPC Anda.
Simpan file di Amazon S3 setelah menyalin ke Amazon Redshift	Aktifkan opsi ini untuk menyimpan file yang diekstrak di Amazon S3 setelah menyalin file-file ini ke Amazon Redshift.

Gunakan petunjuk dalam tabel berikut untuk memberikan informasi untuk pengaturan Salin.

Untuk parameter ini	Lakukan hal berikut
Jumlah kesalahan maksimum	Masukkan jumlah kesalahan pemuatan. Setelah operasi mencapai batas ini, agen ekstraksi AWS SCT data mengakhiri proses pemuatan data. Nilai default adalah 0, yang berarti bahwa agen ekstraksi AWS SCT data melanjutkan beban data terlepas dari kegagalan.

Untuk parameter ini	Lakukan hal berikut
Ganti karakter UTF-8 yang tidak valid	Aktifkan opsi ini untuk mengganti karakter UTF-8 yang tidak valid dengan karakter yang ditentukan dan lanjutkan operasi pemuatan data.
Gunakan kosong sebagai nilai nol	Aktifkan opsi ini untuk memuat bidang kosong yang terdiri dari karakter spasi putih sebagai nol.
Gunakan kosong sebagai nilai null	Aktifkan opsi ini untuk memuat kosong CHAR dan VARCHAR bidang sebagai null.
Memotong kolom	Aktifkan opsi ini untuk memotong data dalam kolom agar sesuai dengan spesifikasi tipe data.
Kompresi otomatis	Aktifkan opsi ini untuk menerapkan pengkodean kompresi selama operasi penyalinan.
Penyegaran statistik otomatis	Aktifkan opsi ini untuk menyegarkan statistik di akhir operasi penyalinan.
Periksa file sebelum memuat	Aktifkan opsi ini untuk memvalidasi file data sebelum memuatnya ke Amazon Redshift.

Menyortir data sebelum bermigrasi menggunakan AWS SCT

Menyortir data Anda sebelum migrasi dengan AWS SCT memberikan beberapa manfaat. Jika Anda mengurutkan data terlebih dahulu, AWS SCT dapat memulai ulang agen ekstraksi pada titik terakhir yang disimpan setelah kegagalan. Selain itu, jika Anda memigrasikan data ke Amazon Redshift dan mengurutkan data terlebih dahulu AWS SCT, dapat menyisipkan data ke Amazon Redshift lebih cepat.

Manfaat ini berkaitan dengan cara AWS SCT membuat kueri ekstraksi data. Dalam beberapa kasus, AWS SCT menggunakan fungsi analitik DENSE_RANK dalam kueri ini. Namun, DENSE_RANK dapat menggunakan banyak waktu dan sumber daya server untuk mengurutkan kumpulan data yang dihasilkan dari ekstraksi, jadi jika AWS SCT dapat bekerja tanpanya, itu berhasil.

Untuk mengurutkan data sebelum bermigrasi menggunakan AWS SCT

1. Buka AWS SCT proyek.
2. Buka menu konteks (klik kanan) untuk objek, lalu pilih Buat tugas Lokal.
3. Pilih tab Advanced, dan untuk strategi Sorting, pilih opsi:
 - Jangan pernah menggunakan penyortiran — Agen ekstraksi tidak menggunakan fungsi analitik DENSE_RANK dan memulai ulang dari awal jika terjadi kegagalan.
 - Gunakan pengurutan jika memungkinkan — Agen ekstraksi menggunakan DENSE_RANK jika tabel memiliki kunci utama atau kendala unik.
 - Gunakan penyortiran setelah gagal pertama (disarankan) - Agen ekstraksi pertama mencoba mendapatkan data tanpa menggunakan DENSE_RANK. Jika upaya pertama gagal, agen ekstraksi membangun kembali kueri menggunakan DENSE_RANK dan mempertahankan lokasinya jika terjadi kegagalan.

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

4. Tetapkan parameter tambahan seperti yang dijelaskan berikut, lalu pilih Buat untuk membuat tugas ekstraksi data Anda.

Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data

Gunakan prosedur berikut untuk membuat, menjalankan, dan memantau tugas ekstraksi data.

Untuk menetapkan tugas ke agen dan memigrasikan data

1. Dalam AWS Schema Conversion Tool, setelah Anda mengonversi skema Anda, pilih satu atau beberapa tabel dari panel kiri proyek Anda.

Anda dapat memilih semua tabel, tetapi kami merekomendasikan untuk tidak melakukannya karena alasan kinerja. Kami menyarankan Anda membuat beberapa tugas untuk beberapa tabel berdasarkan ukuran tabel di gudang data Anda.

2. Buka menu konteks (klik kanan) untuk setiap tabel, lalu pilih Buat tugas. Kotak dialog Create Local task terbuka.
3. Untuk nama Tugas, masukkan nama untuk tugas tersebut.
4. Untuk mode Migrasi, pilih salah satu dari berikut ini:
 - Ekstrak saja - Ekstrak data Anda, dan simpan data ke folder kerja lokal Anda.
 - Ekstrak dan unggah - Ekstrak data Anda, dan unggah data Anda ke Amazon S3.
 - Ekstrak, unggah, dan salin — Ekstrak data Anda, unggah data Anda ke Amazon S3, dan salin ke gudang data Amazon Redshift Anda.
5. Untuk jenis Enkripsi, pilih salah satu dari berikut ini:
 - NONE — Matikan enkripsi data untuk seluruh proses migrasi data.
 - CSE_SK — Gunakan enkripsi sisi klien dengan kunci simetris untuk memigrasikan data. AWS SCT secara otomatis menghasilkan kunci enkripsi dan mengirimkannya ke agen ekstraksi data menggunakan Secure Sockets Layer (SSL). AWS SCT tidak mengenkripsi objek besar (LOB) selama migrasi data.
6. Pilih Ekstrak LOB untuk mengekstrak benda besar. Jika Anda tidak perlu mengekstrak benda besar, Anda dapat menghapus kotak centang. Melakukan hal ini mengurangi jumlah data yang Anda ekstrak.
7. Untuk melihat informasi terperinci tentang tugas, pilih Aktifkan pencatatan tugas. Anda dapat menggunakan log tugas untuk men-debug masalah.

Jika Anda mengaktifkan pencatatan tugas, pilih tingkat detail yang ingin Anda lihat. Levelnya adalah sebagai berikut, dengan setiap level termasuk semua pesan dari level sebelumnya:

- ERROR— Jumlah detail terkecil.
- WARNING
- INFO

- DEBUG
 - TRACE— Jumlah detail terbesar.
8. Untuk mengeksport data dari BigQuery, AWS SCT gunakan folder bucket Google Cloud Storage. Dalam folder ini, agen ekstraksi data menyimpan data sumber Anda.

Untuk memasukkan jalur ke folder bucket Google Cloud Storage Anda, pilih Advanced. Untuk folder bucket Google CS, masukkan nama bucket dan nama folder.
 9. Untuk mengambil peran bagi pengguna agen ekstraksi data Anda, pilih pengaturan Amazon S3. Untuk peran IAM, masukkan nama peran yang akan digunakan. Untuk Wilayah, pilih Wilayah AWS untuk peran ini.
 10. Pilih Tugas uji untuk memverifikasi bahwa Anda dapat terhubung ke folder kerja, bucket Amazon S3, dan gudang data Amazon Redshift. Verifikasi tergantung pada mode migrasi yang Anda pilih.
 11. Memilih Membuat untuk membuat tugas.
 12. Ulangi langkah sebelumnya untuk membuat tugas untuk semua data yang ingin Anda migrasi.

Untuk menjalankan dan memantau tugas

1. Untuk Tampilan, pilih tampilan Migrasi Data. Tab Agen muncul.
2. Pilih tab Tugas. Tugas Anda muncul di kisi di bagian atas seperti yang ditunjukkan berikut. Anda dapat melihat status tugas di kisi atas, dan status subtugasnya di kisi bawah.

The screenshot displays the AWS SCT interface with the 'Tasks' tab selected. The main area shows a table of migration tasks with columns for Name, Extract, Upload, and Copy. Each task has a progress bar and a status icon (green checkmark for success, yellow play button for pending, or red X for failure).

Name	Extract	Upload	Copy
+ CUSTOMER	+ 0%		
+ LINEORDER_100K	+ 0%		
+ LINEORDER_150K	+ 0%		
+ LINEORDER_1M	+ 0%		
LocalTask 2	100%	100%	
+ CUSTOMER	100%	100%	
+ LINEORDER_100K	100%	100%	
+ LINEORDER_150K	100%	100%	
LocalTask 3	100%	100%	0%
+ LINEORDER_100K	100%	100%	0%

Below the table, there are control buttons: Download log, All migration Tasks, Resume, Stop, Restart, Reset, Delete, Replace, Refresh all, and Refresh selected. At the bottom, the 'Processing details' tab is active, showing 'Extracting' and 'Data' sections with a progress bar at 100%.

3. Pilih tugas di grid atas dan perluas. Bergantung pada mode migrasi yang Anda pilih, Anda melihat tugas dibagi menjadi Ekstrak, Unggah, dan Salin.
4. Pilih Mulai untuk tugas untuk memulai tugas itu. Anda dapat memantau status tugas Anda saat mereka bekerja. Subtugas berjalan secara paralel. Ekstrak, unggah, dan salin juga berjalan secara paralel.
5. Jika Anda mengaktifkan logging saat mengatur tugas, Anda dapat melihat log:
 - a. Pilih Unduh log. Sebuah pesan muncul dengan nama folder yang berisi file log. Singkirkan pesannya.
 - b. Tautan muncul di tab Detail tugas. Pilih tautan untuk membuka folder yang berisi file log.

Anda dapat menutup AWS SCT, dan agen serta tugas Anda terus berjalan. Anda dapat membuka kembali AWS SCT nanti untuk memeriksa status tugas Anda dan melihat log tugas.

Anda dapat menyimpan tugas ekstraksi data ke disk lokal Anda dan mengembalikannya ke proyek yang sama atau lainnya dengan menggunakan ekspor dan impor. Untuk mengekspor tugas, pastikan Anda memiliki setidaknya satu tugas ekstraksi yang dibuat dalam sebuah proyek. Anda dapat mengimpor tugas ekstraksi tunggal atau semua tugas yang dibuat dalam proyek.

Saat Anda mengekspor tugas ekstraksi, AWS SCT buat `.xml` file terpisah untuk tugas itu. `.xml`File menyimpan informasi metadata tugas tersebut, seperti properti tugas, deskripsi, dan subtugas. `.xml`File tidak berisi informasi tentang pemrosesan tugas ekstraksi. Informasi seperti berikut ini dibuat ulang saat tugas diimpor:

- Kemajuan tugas
- Subtugas dan status panggung
- Distribusi agen ekstraksi berdasarkan subtugas dan tahapan
- ID tugas dan subtugas
- Nama tugas

Mengekspor dan mengimpor tugas ekstraksi AWS SCT data

Anda dapat dengan cepat menyimpan tugas yang ada dari satu proyek dan mengembalikannya di proyek lain (atau proyek yang sama) menggunakan AWS SCT ekspor dan impor. Gunakan prosedur berikut untuk mengekspor dan mengimpor tugas ekstraksi data.

Untuk mengekspor dan mengimpor tugas ekstraksi data

1. Untuk Tampilan, pilih tampilan Migrasi Data. Tab Agen muncul.
2. Pilih tab Tugas. Tugas Anda tercantum dalam kisi yang muncul.
3. Pilih tiga titik yang disejajarkan secara vertikal (ikon elipsis) yang terletak di sudut kanan bawah di bawah daftar tugas.
4. Pilih tugas Ekspor dari menu pop-up.
5. Pilih folder tempat Anda AWS SCT ingin menempatkan .xml file ekspor tugas.

AWS SCT membuat file ekspor tugas dengan format nama file *TASK-DESCRIPTION_TASK-ID.xml*.

6. Pilih tiga titik yang disejajarkan secara vertikal (ikon elipsis) di kanan bawah di bawah daftar tugas.
7. Pilih Impor tugas dari menu pop-up.

Anda dapat mengimpor tugas ekstraksi ke proyek yang terhubung ke database sumber, dan proyek memiliki setidaknya satu agen ekstraksi terdaftar aktif.

8. Pilih .xml file untuk tugas ekstraksi yang Anda ekspor.

AWS SCT mendapatkan parameter tugas ekstraksi dari file, membuat tugas, dan menambahkan tugas ke agen ekstraksi.

9. Ulangi langkah-langkah ini untuk mengekspor dan mengimpor tugas ekstraksi data tambahan.

Pada akhir proses ini, ekspor dan impor Anda selesai dan tugas ekstraksi data Anda siap digunakan.

Ekstraksi data menggunakan perangkat AWS Snowball Edge

Proses penggunaan AWS SCT dan AWS Snowball Edge memiliki beberapa langkah. Migrasi melibatkan tugas lokal, di mana AWS SCT menggunakan agen ekstraksi data untuk memindahkan data ke perangkat AWS Snowball Edge, lalu tindakan perantara di mana AWS menyalin data dari perangkat AWS Snowball Edge ke bucket Amazon S3. Proses selesai AWS SCT memuat data dari bucket Amazon S3 ke Amazon Redshift.

Bagian berikut ikhtisar ini memberikan step-by-step panduan untuk masing-masing tugas ini. Prosedur ini mengasumsikan bahwa Anda telah AWS SCT instal dan bahwa Anda telah mengonfigurasi dan mendaftarkan agen ekstraksi data pada mesin khusus.

Lakukan langkah-langkah berikut untuk memigrasikan data dari penyimpanan data lokal ke penyimpanan AWS data menggunakan AWS Snowball Edge.

1. Buat pekerjaan AWS Snowball Edge menggunakan AWS Snowball konsol.
2. Buka kunci perangkat AWS Snowball Edge menggunakan mesin Linux lokal khusus.
3. Buat proyek baru di AWS SCT.
4. Instal dan konfigurasi agen ekstraksi data Anda.
5. Buat dan setel izin untuk bucket Amazon S3 untuk digunakan.
6. Impor AWS Snowball pekerjaan ke AWS SCT proyek Anda.
7. Daftarkan agen ekstraksi data Anda di AWS SCT.
8. Buat tugas lokal di AWS SCT.
9. Jalankan dan pantau tugas migrasi data di AWS SCT.

tep-by-step Prosedur S untuk memigrasi data menggunakan AWS SCT dan AWS Snowball Edge

Bagian berikut memberikan informasi rinci tentang langkah-langkah migrasi.

Langkah 1: Buat pekerjaan AWS Snowball Edge

Buat AWS Snowball pekerjaan dengan mengikuti langkah-langkah yang diuraikan di bagian [Membuat Pekerjaan AWS Snowball Tepi](#) di Panduan Pengembang AWS Snowball Edge.

Langkah 2: Buka kunci perangkat AWS Snowball Edge

Jalankan perintah yang membuka kunci dan memberikan kredensial ke perangkat Snowball Edge dari mesin tempat Anda menginstal agen. AWS DMS Dengan menjalankan perintah ini, Anda dapat yakin bahwa panggilan AWS DMS agen terhubung ke perangkat AWS Snowball Edge. Untuk informasi selengkapnya tentang membuka kunci perangkat AWS Snowball Edge, lihat [Membuka Kunci Snowball](#) Edge.

```
aws s3 ls s3://<bucket-name> --profile <Snowball Edge profile> --endpoint http://<Snowball IP>:8080 --recursive
```

Langkah 3: Buat AWS SCT proyek baru

Selanjutnya, buat AWS SCT proyek baru.

Untuk membuat proyek baru di AWS SCT

1. Mulai AWS Schema Conversion Tool. Pada menu File, pilih Proyek baru. Kotak dialog Proyek baru muncul.
2. Memasukkan nama untuk proyek Anda, yang menyimpan secara lokal di komputer Anda.
3. Memasukkan lokasi untuk file proyek lokal Anda.
4. Pilih OK untuk membuat AWS SCT proyek Anda.
5. Pilih Tambahkan sumber untuk menambahkan database sumber baru ke AWS SCT proyek Anda.
6. Pilih Tambahkan target untuk menambahkan platform target baru di AWS SCT proyek Anda.
7. Pilih skema basis data sumber di panel kiri.
8. Di panel kanan, tentukan platform basis data target untuk skema sumber yang dipilih.
9. Pilih Buat pemetaan. Tombol ini menjadi aktif setelah Anda memilih skema basis data sumber dan platform basis data target.

Langkah 4: Instal dan konfigurasi agen ekstraksi data Anda

AWS SCT menggunakan agen ekstraksi data untuk memigrasikan data ke Amazon Redshift. File.zip yang Anda unduh untuk diinstal AWS SCT, termasuk file penginstal agen ekstraksi. Anda dapat menginstal agen ekstraksi data di Windows, Red Hat Enterprise Linux, atau Ubuntu. Untuk informasi selengkapnya, lihat [Instalasi agen ekstraksi](#).

Untuk mengonfigurasi agen ekstraksi data Anda, masukkan mesin basis data sumber dan target Anda. Selain itu, pastikan Anda mengunduh driver JDBC untuk basis data sumber dan target Anda di komputer tempat Anda menjalankan agen ekstraksi data Anda. Agen ekstraksi data menggunakan driver ini untuk terhubung ke basis data sumber dan target Anda. Untuk informasi selengkapnya, lihat [Mengunduh driver database yang diperlukan](#).

Di Windows, penginstal agen ekstraksi data meluncurkan wizard konfigurasi di jendela prompt perintah. Di Linux, jalankan `sct-extractor-setup.sh` file dari lokasi tempat Anda menginstal agen.

Langkah 5: Konfigurasi AWS SCT untuk mengakses bucket Amazon S3

Untuk informasi tentang mengonfigurasi bucket Amazon S3, [lihat ikhtisar Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#).

Langkah 6: Impor AWS Snowball pekerjaan ke AWS SCT proyek Anda

Untuk menghubungkan AWS SCT proyek Anda dengan perangkat AWS Snowball Edge Anda, impor AWS Snowball pekerjaan Anda.

Untuk mengimpor AWS Snowball pekerjaan Anda

1. Buka menu Pengaturan, lalu pilih Pengaturan global. Kotak dialog Pengaturan global muncul.
2. Pilih profil AWS layanan, lalu pilih Impor pekerjaan.
3. Pilih AWS Snowball pekerjaan Anda.
4. Masukkan AWS Snowball IP Anda. Untuk informasi selengkapnya, lihat [Mengubah Alamat IP Anda](#) di Panduan AWS Snowball Pengguna.
5. Masukkan AWS Snowball Port Anda. Untuk informasi selengkapnya, lihat [Port yang Diperlukan untuk Menggunakan AWS Layanan di Perangkat AWS Snowball Edge](#) di Panduan Pengembang AWS Snowball Edge.
6. Masukkan kunci AWS Snowball akses dan kunci AWS Snowball rahasia Anda. Untuk informasi selengkapnya, lihat [Otorisasi dan Kontrol Akses AWS Snowball di](#) Panduan AWS Snowball Pengguna.
7. Pilih Berlakukan, lalu pilih OKE.

Langkah 7: Daftarkan agen ekstraksi data di AWS SCT

Di bagian ini, Anda mendaftarkan agen ekstraksi data di AWS SCT.

Untuk mendaftarkan agen ekstraksi data

1. Pada menu Tampilan, pilih Tampilan migrasi data (lainnya), lalu pilih Daftar.
2. Untuk Deskripsi, masukkan nama untuk agen ekstraksi data Anda.
3. Untuk nama Host, masukkan alamat IP komputer tempat Anda menjalankan agen ekstraksi data Anda.
4. Untuk Port, masukkan port mendengarkan yang Anda konfigurasi.
5. Pilih Pendaftaran.

Langkah 8: Membuat tugas lokal

Selanjutnya, Anda membuat tugas migrasi. Tugas tersebut mencakup dua subtugas. Satu subtugas memigrasikan data dari database sumber ke alat AWS Snowball Edge. Subtugas lainnya mengambil data yang dimuat alat ke dalam bucket Amazon S3 dan memigrasikannya ke database target.

Untuk membuat tugas migrasi

1. Pada menu Tampilan, lalu pilih Tampilan migrasi data (lainnya).
2. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih objek skema untuk dimigrasi. Buka menu konteks (klik kanan) untuk objek, lalu pilih Buat tugas lokal.
3. Untuk nama Tugas, masukkan nama deskriptif untuk tugas migrasi data Anda.
4. Untuk mode Migrasi, pilih Ekstrak, unggah, dan salin.
5. Pilih pengaturan Amazon S3.
6. Pilih Gunakan Snowball.
7. Masukkan folder dan subfolder di bucket Amazon S3 tempat agen ekstraksi data dapat menyimpan data.
8. Memilih Membuat untuk membuat tugas.

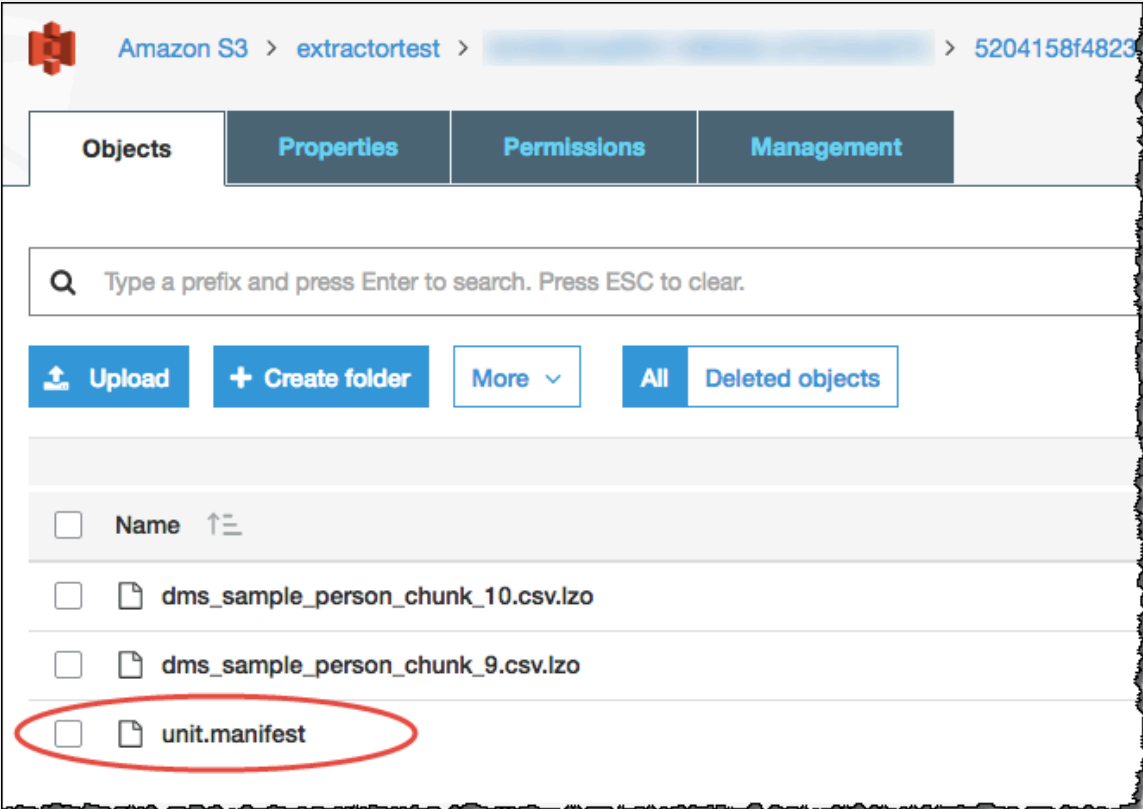
Langkah 9: Menjalankan dan memantau tugas migrasi data di AWS SCT

Untuk memulai tugas migrasi data, pilih Mulai. Pastikan Anda membuat koneksi ke database sumber, bucket Amazon S3, AWS Snowball perangkat, serta koneksi ke database target aktif. AWS

Anda dapat memantau dan mengelola tugas migrasi data dan subtugasnya di tab Tugas. Anda dapat melihat kemajuan migrasi data, serta menjeda atau memulai ulang tugas migrasi data Anda.

Output tugas ekstraksi data

Setelah tugas migrasi Anda selesai, data Anda siap. Gunakan informasi berikut untuk menentukan cara melanjutkan berdasarkan mode migrasi yang Anda pilih dan lokasi data Anda.

Modus migrasi	Lokasi data																
Ekstrak, unggah, dan salin	Data sudah ada di gudang data Amazon Redshift Anda. Anda dapat memverifikasi bahwa data ada di sana, dan mulai menggunakannya. Untuk informasi selengkapnya, lihat Menghubungkan ke klaster dari alat dan kode klien .																
Ekstrak dan unggah	<p>Agan ekstraksi menyimpan data Anda sebagai file di bucket Amazon S3 Anda. Anda dapat menggunakan perintah Amazon Redshift COPY untuk memuat data Anda ke Amazon Redshift. Untuk informasi selengkapnya, lihat Memuat data dari Amazon S3 di dokumentasi Amazon Redshift.</p> <p>Ada beberapa folder di bucket Amazon S3 Anda, sesuai dengan tugas ekstraksi yang Anda siapkan. Saat memuat data ke Amazon Redshift, tentukan nama file manifes yang dibuat oleh setiap tugas. File manifes muncul di folder tugas di bucket Amazon S3 Anda seperti yang ditunjukkan berikut.</p>  <p>The screenshot shows the Amazon S3 console interface for a bucket named 'extractortest'. The breadcrumb navigation 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'. The list of objects is as follows:</p> <table border="1"><thead><tr><th><input type="checkbox"/></th><th>Name</th><th>↑</th><th>☰</th></tr></thead><tbody><tr><td><input type="checkbox"/></td><td>dms_sample_person_chunk_10.csv.lzo</td><td></td><td></td></tr><tr><td><input type="checkbox"/></td><td>dms_sample_person_chunk_9.csv.lzo</td><td></td><td></td></tr><tr><td><input type="checkbox"/></td><td>unit.manifest</td><td></td><td></td></tr></tbody></table> <p>The 'unit.manifest' file is circled in red in the original image.</p>	<input type="checkbox"/>	Name	↑	☰	<input type="checkbox"/>	dms_sample_person_chunk_10.csv.lzo			<input type="checkbox"/>	dms_sample_person_chunk_9.csv.lzo			<input type="checkbox"/>	unit.manifest		
<input type="checkbox"/>	Name	↑	☰														
<input type="checkbox"/>	dms_sample_person_chunk_10.csv.lzo																
<input type="checkbox"/>	dms_sample_person_chunk_9.csv.lzo																
<input type="checkbox"/>	unit.manifest																

Modus migrasi	Lokasi data
Ekstrak saja	Agen ekstraksi menyimpan data Anda sebagai file di folder kerja Anda. Salin data Anda secara manual ke bucket Amazon S3 Anda, lalu lanjutkan dengan instruksi untuk Ekstrak dan unggah.

Menggunakan partisi virtual dengan AWS Schema Conversion Tool

Anda sering dapat mengelola tabel non-partisi besar dengan membuat subtugas yang membuat partisi virtual dari data tabel menggunakan aturan pemfilteran. Di AWS SCT, Anda dapat membuat partisi virtual untuk data yang dimigrasi. Ada tiga jenis partisi, yang bekerja dengan tipe data tertentu:

- Jenis partisi RANGE bekerja dengan tipe data numerik dan tanggal dan waktu.
- Jenis partisi LIST bekerja dengan tipe data numerik, karakter, dan tanggal dan waktu.
- Jenis partisi DATE AUTO SPLIT bekerja dengan tipe data numerik, tanggal, dan waktu.

AWS SCT memvalidasi nilai yang Anda berikan untuk membuat partisi. Misalnya, jika Anda mencoba mempartisi kolom dengan tipe data NUMERIC tetapi Anda memberikan nilai dari tipe data yang berbeda, akan AWS SCT menimbulkan kesalahan.

Selain itu, jika Anda menggunakan AWS SCT untuk memigrasikan data ke Amazon Redshift, Anda dapat menggunakan partisi asli untuk mengelola migrasi tabel besar. Untuk informasi selengkapnya, lihat [Menggunakan partisi asli](#).

Batas saat membuat partisi virtual

Ini adalah batasan untuk membuat partisi virtual:

- Anda hanya dapat menggunakan partisi virtual hanya untuk tabel yang tidak dipartisi.
- Anda dapat menggunakan partisi virtual hanya dalam tampilan migrasi data.
- Anda tidak dapat menggunakan opsi UNION ALL VIEW dengan partisi virtual.

Jenis partisi RANGE

Jenis partisi RANGE partisi data berdasarkan rentang nilai kolom untuk tipe data numerik dan tanggal dan waktu. Jenis partisi ini membuat WHERE klausa, dan Anda memberikan rentang nilai untuk setiap partisi. Untuk menentukan daftar nilai untuk kolom yang dipartisi, gunakan kotak Nilai. Anda dapat memuat informasi nilai dengan menggunakan file.csv.

Jenis partisi RANGE menciptakan partisi default di kedua ujung nilai partisi. Partisi default ini menangkap data apa pun yang kurang dari atau lebih besar dari nilai partisi yang ditentukan.

Misalnya, Anda dapat membuat beberapa partisi berdasarkan rentang nilai yang Anda berikan. Dalam contoh berikut, nilai partisi untuk LO_TAX ditentukan untuk membuat beberapa partisi.

```
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
```

Untuk membuat partisi virtual RANGE

1. Terbuka AWS SCT.
2. Pilih mode tampilan Migrasi Data (lainnya).
3. Pilih tabel tempat Anda ingin mengatur partisi virtual. Buka menu konteks (klik kanan) untuk tabel, dan pilih Tambahkan partisi virtual.
4. Dalam Tambahkan partisi virtual kotak dialog, masukkan informasi sebagai berikut.

Opsi	Tindakan
Jenis partisi	Pilih RANGE. UI kotak dialog berubah tergantung pada jenis yang Anda pilih.
Nama kolom	Pilih kolom yang ingin Anda partisi.
Jenis kolom	Pilih tipe data untuk nilai di kolom.
Nilai-nilai	Tambahkan nilai baru dengan mengetikkan setiap nilai di kotak Nilai Baru, lalu pilih tanda plus untuk menambahkan nilai.
Muat dari file	(Opsional) Masukkan nama file.csv yang berisi nilai partisi.

5. Pilih OK.

LIST jenis partisi

Jenis partisi LIST data partisi berdasarkan nilai kolom untuk tipe data numerik, karakter, dan tanggal dan waktu. Jenis partisi ini membuat WHERE klausa, dan Anda memberikan nilai untuk setiap partisi. Untuk menentukan daftar nilai untuk kolom yang dipartisi, gunakan kotak Nilai. Anda dapat memuat informasi nilai dengan menggunakan file.csv.

Misalnya, Anda dapat membuat beberapa partisi berdasarkan nilai yang Anda berikan. Dalam contoh berikut, nilai partisi untuk LO_ORDERKEY ditentukan untuk membuat beberapa partisi.

```
Partition1: WHERE LO_ORDERKEY = 1
Partition2: WHERE LO_ORDERKEY = 2
Partition3: WHERE LO_ORDERKEY = 3
...
PartitionN: WHERE LO_ORDERKEY = USER_VALUE_N
```

Anda juga dapat membuat partisi default untuk nilai-nilai yang tidak termasuk dalam yang ditentukan.

Anda dapat menggunakan jenis partisi LIST untuk memfilter data sumber jika Anda ingin mengecualikan nilai tertentu dari migrasi. Misalnya, anggaplah Anda ingin menghilangkan baris dengan LO_ORDERKEY = 4. Dalam hal ini, jangan sertakan nilai 4 dalam daftar nilai partisi dan pastikan bahwa Sertakan nilai lain tidak dipilih.

Untuk membuat partisi virtual LIST

1. Terbuka AWS SCT.
2. Pilih mode tampilan Migrasi Data (lainnya).
3. Pilih tabel tempat Anda ingin mengatur partisi virtual. Buka menu konteks (klik kanan) untuk tabel, dan pilih Tambahkan partisi virtual.
4. Dalam Tambahkan partisi virtual kotak dialog, masukkan informasi sebagai berikut.

Opsi	Tindakan
Jenis partisi	Pilih DAFTAR. UI kotak dialog berubah tergantung pada jenis yang Anda pilih.

Opsi	Tindakan
Nama kolom	Pilih kolom yang ingin Anda partisi.
Nilai baru	Ketik nilai di sini untuk menambahkannya ke set nilai partisi.
Sertakan nilai lainnya	Pilih opsi ini untuk membuat partisi default di mana semua nilai yang tidak memenuhi kriteria partisi disimpan.
Muat dari file	(Opsional) Masukkan nama file.csv yang berisi nilai partisi.

5. Pilih OK.

Jenis partisi DATE AUTO SPLIT

Jenis partisi DATE AUTO SPLIT adalah cara otomatis untuk menghasilkan partisi RANGE. Dengan DATA AUTO SPLIT, Anda memberi tahu AWS SCT atribut partisi, di mana harus memulai dan mengakhiri, dan ukuran rentang antara nilai-nilai. Kemudian AWS SCT menghitung nilai partisi secara otomatis.

DATA AUTO SPLIT mengotomatiskan banyak pekerjaan yang terlibat dengan pembuatan partisi rentang. Tradeoff antara menggunakan teknik ini dan partisi rentang adalah seberapa banyak kontrol yang Anda butuhkan atas batas partisi. Proses pemisahan otomatis selalu menciptakan rentang ukuran (seragam) yang sama. Partisi rentang memungkinkan Anda untuk memvariasikan ukuran setiap rentang sesuai kebutuhan untuk distribusi data tertentu Anda. Misalnya, Anda dapat menggunakan harian, mingguan, dua mingguan, bulanan, dan sebagainya.

```
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'
```

Untuk membuat partisi virtual DATE AUTO SPLIT

1. Terbuka AWS SCT.
2. Pilih mode tampilan Migrasi Data (lainnya).
3. Pilih tabel tempat Anda ingin mengatur partisi virtual. Buka menu konteks (klik kanan) untuk tabel, dan pilih Tambahkan partisi virtual.

4. Dalam Tambahkan partisi virtual kotak dialog, masukkan informasi sebagai berikut.

Opsi	Tindakan
Jenis partisi	Pilih DATE AUTO SPLIT. UI kotak dialog berubah tergantung pada jenis yang Anda pilih.
Nama kolom	Pilih kolom yang ingin Anda partisi.
Tanggal mulai	Ketik tanggal mulai.
Tanggal akhir	Ketik tanggal akhir.
Interval	Masukkan unit interval, dan pilih nilai untuk unit itu.

5. Pilih OK.

Menggunakan partisi asli

Untuk mempercepat migrasi data, agen ekstraksi data Anda dapat menggunakan partisi asli tabel di server gudang data sumber Anda. AWS SCT mendukung partisi asli untuk migrasi dari Greenplum, Netezza, dan Oracle ke Amazon Redshift.

Misalnya, setelah membuat proyek, Anda dapat mengumpulkan statistik pada skema dan menganalisis ukuran tabel yang dipilih untuk migrasi. Untuk tabel yang melebihi ukuran yang ditentukan, AWS SCT memicu mekanisme partisi asli.

Untuk menggunakan partisi asli

1. Buka AWS SCT, dan pilih Proyek baru untuk File. Kotak dialog Proyek baru muncul.
2. Buat proyek baru, tambahkan server sumber dan target Anda, dan buat aturan pemetaan. Untuk informasi selengkapnya, lihat [MembuatAWS SCT proyek](#).
3. Pilih Tampilan, lalu pilih Tampilan utama.
4. Untuk pengaturan Project, pilih tab Migrasi data. Pilih Gunakan partisi otomatis. Untuk database sumber Greenplum dan Netezza, masukkan ukuran minimal tabel yang didukung dalam megabyte (misalnya, 100). AWS SCT secara otomatis membuat subtugas migrasi terpisah untuk setiap partisi asli yang tidak kosong. Untuk migrasi Oracle ke Amazon Redshift, buat subtugas untuk semua tabel AWS SCT yang dipartisi.

5. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih skema. Buka menu konteks (klik kanan) untuk objek, dan pilih Kumpulkan statistik. Untuk migrasi data dari Oracle ke Amazon Redshift, Anda dapat melewati langkah ini.
6. Pilih semua tabel untuk dimigrasi.
7. Daftarkan jumlah agen yang diperlukan. Untuk informasi selengkapnya, lihat [Mendaftarkan agen ekstraksi dengan AWS Schema Conversion Tool](#).
8. Buat tugas ekstraksi data untuk tabel yang dipilih. Untuk informasi selengkapnya, lihat [Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data](#).

Periksa apakah tabel besar dibagi menjadi subtugas, dan setiap subtugas cocok dengan kumpulan data yang menyajikan bagian tabel yang terletak pada satu irisan di gudang data sumber Anda.

9. Mulai dan pantau proses migrasi hingga agen ekstraksi AWS SCT data menyelesaikan migrasi data dari tabel sumber Anda.

Migrasi LOB ke Amazon Redshift

Amazon Redshift tidak mendukung penyimpanan objek biner besar (LOB). Namun, jika Anda perlu memigrasikan satu atau beberapa LOB ke Amazon Redshift AWS SCT, dapat melakukan migrasi. Untuk melakukannya, AWS SCT gunakan bucket Amazon S3 untuk menyimpan LOB dan menulis URL untuk bucket Amazon S3 ke dalam data migrasi yang disimpan di Amazon Redshift.

Untuk memigrasikan LOB ke Amazon Redshift

1. Buka AWS SCT proyek.
2. Connect ke database sumber dan target. Segarkan metadata dari database target, dan pastikan tabel yang dikonversi ada di sana.
3. Untuk Tindakan, pilih Buat tugas lokal.
4. Untuk mode Migrasi, pilih salah satu dari berikut ini:
 - Ekstrak dan unggah untuk mengekstrak data Anda, dan unggah data Anda ke Amazon S3.
 - Ekstrak, unggah, dan salin untuk mengekstrak data Anda, unggah data Anda ke Amazon S3, dan salin ke gudang data Amazon Redshift Anda.
5. Pilih pengaturan Amazon S3.

6. Untuk folder LOB bucket Amazon S3, masukkan nama folder di bucket Amazon S3 tempat Anda ingin LOB disimpan.

Jika Anda menggunakan profil AWS layanan, bidang ini bersifat opsional. AWS SCT dapat menggunakan pengaturan default dari profil Anda. Untuk menggunakan bucket Amazon S3 lainnya, masukkan jalur di sini.

7. Aktifkan opsi Gunakan proxy untuk menggunakan server proxy untuk mengunggah data ke Amazon S3. Kemudian pilih protokol transfer data, masukkan nama host, port, nama pengguna, dan kata sandi.
8. Untuk tipe Endpoint, pilih FIPS untuk menggunakan titik akhir Federal Information Processing Standard (FIPS). Pilih VPCE untuk menggunakan titik akhir virtual private cloud (VPC). Kemudian untuk titik akhir VPC, masukkan Domain Name System (DNS) dari titik akhir VPC Anda.
9. Aktifkan Simpan file di Amazon S3 setelah menyalin ke Amazon Redshift opsi untuk menyimpan file yang diekstraksi di Amazon S3 setelah menyalin file-file ini ke Amazon Redshift.
10. Memilih Membuat untuk membuat tugas.

Praktik terbaik dan pemecahan masalah untuk agen ekstraksi data

Berikut ini adalah beberapa praktik terbaik dan saran pemecahan masalah untuk menggunakan agen ekstraksi.

Masalah	Saran pemecahan masalah
Kinerja lambat	Untuk meningkatkan kinerja, kami merekomendasikan yang berikut: <ul style="list-style-type: none">• Instal beberapa agen.• Instal agen di komputer yang dekat dengan gudang data Anda.• Jangan menjalankan semua tabel pada satu tugas agen.
Penundaan pertikaian	Hindari terlalu banyak agen yang mengakses gudang data Anda secara bersamaan.

Masalah	Saran pemecahan masalah
Seorang agen turun sementara	<p>Jika agen down, status setiap tugasnya muncul sebagai gagal dalam AWS SCT. Jika Anda menunggu, dalam beberapa kasus agen dapat pulih. Dalam hal ini, status tugasnya diperbarui di AWS SCT.</p>
Seorang agen turun secara permanen	<p>Jika komputer yang menjalankan agen mati secara permanen, dan agen tersebut menjalankan tugas, Anda dapat mengganti agen baru untuk melanjutkan tugas. Anda dapat mengganti agen baru hanya jika folder kerja agen asli tidak berada di komputer yang sama dengan agen asli. Untuk mengganti agen baru, lakukan hal berikut:</p> <ul style="list-style-type: none">• Instal agen di komputer baru.• Konfigurasi agen baru dengan pengaturan yang sama, termasuk nomor port dan folder kerja, seperti agen asli.• Mulai agen. Setelah agen dimulai, tugas menemukan agen baru yang tersedia dan terus berjalan pada agen baru.

Mengonversi aplikasi SQL menggunakan AWS SCT

Ketika Anda mengonversi skema database Anda dari satu mesin ke mesin lain, Anda juga perlu memperbarui kode SQL dalam aplikasi Anda untuk berinteraksi dengan mesin database baru, bukan yang lama. Anda dapat melihat, menganalisis, dan menyimpan kode SQL yang telah dikonversi.

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi kode SQL di C++, C#, Java, atau kode aplikasi lainnya. Untuk konversi Oracle ke PostgreSQL, Anda dapat menggunakan AWS SCT untuk mengonversi kode SQL*Plus ke PSQL. Juga, untuk konversi Oracle ke PostgreSQL, Anda dapat menggunakan AWS SCT untuk mengonversi kode SQL yang disematkan ke dalam aplikasi C#, C++, Java, dan Pro*C.

Topik

- [Ikhtisar mengkonversi aplikasi SQL](#)
- [Mengonversi kode SQL dalam aplikasi Anda dengan AWS SCT](#)
- [Mengonversi kode SQL dalam aplikasi C# dengan AWS SCT](#)
- [Mengonversi kode SQL dalam aplikasi C++ dengan AWS SCT](#)
- [Mengonversi kode SQL dalam aplikasi Java dengan AWS SCT](#)
- [Mengonversi kode SQL dalam aplikasi Pro*C dengan AWS SCT](#)

Ikhtisar mengkonversi aplikasi SQL

Untuk mengonversi kode SQL dalam aplikasi, lakukan langkah-langkah tingkat tinggi berikut:

- Buat proyek konversi aplikasi—Proyek konversi aplikasi adalah anak dari proyek konversi skema database. Setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi generik di AWS SCT](#).
- Analisis dan konversi kode SQL—AWS SCT menganalisis aplikasi Anda, mengekstrak kode SQL, dan membuat versi lokal dari SQL yang dikonversi untuk Anda tinjau dan diedit. Alat ini tidak mengubah kode dalam aplikasi Anda sampai Anda siap. Untuk informasi selengkapnya, lihat [Menganalisis dan mengonversi kode SQL Anda AWS SCT](#).
- Buat laporan penilaian aplikasi—Laporan penilaian aplikasi memberikan informasi penting tentang konversi kode SQL aplikasi dari skema basis data sumber Anda ke skema basis data target Anda.

Untuk informasi selengkapnya, lihat [Membuat dan menggunakanAWS SCTlaporan penilaian diAWS SCT](#).

- Edit, terapkan perubahan, dan simpan kode SQL Anda yang telah dikonversi— Laporan penilaian mencakup daftar item kode SQL yang tidak dapat dikonversi secara otomatis. Untuk item ini, Anda dapat mengedit kode SQL secara manual untuk melakukan konversi. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan kode SQL Anda yang dikonversi denganAWS SCT](#).

Mengonversi kode SQL dalam aplikasi Anda denganAWS SCT

Anda dapat menggunakanAWS SCTuntuk mengonversi kode SQL yang disematkan ke dalam aplikasi Anda. GenerikAWS SCTkonverter aplikasi memperlakukan kode aplikasi Anda sebagai teks biasa. Ini memindai kode aplikasi Anda dan mengekstrak kode SQL dengan ekspresi reguler. Konverter ini mendukung berbagai jenis file kode sumber dan bekerja dengan kode aplikasi yang ditulis dalam bahasa pemrograman apa pun.

Konverter aplikasi generik memiliki batasan sebagai berikut. Itu tidak menyelam jauh ke dalam logika aplikasi yang khusus untuk bahasa pemrograman aplikasi Anda. Juga, konverter generik tidak mendukung pernyataan SQL dari objek aplikasi yang berbeda, seperti fungsi, parameter, variabel lokal, dan sebagainya.

Untuk meningkatkan konversi kode SQL aplikasi Anda, gunakan konverter kode SQL aplikasi khusus bahasa. Untuk informasi lebih lanjut, lihat [Mengonversi kode SQL dalam aplikasi C #](#), [Mengonversi kode SQL dalam aplikasi Java](#), dan [Mengonversi kode SQL dalam aplikasi Pro* C](#).

Membuat proyek konversi aplikasi generik diAWS SCT

DalamAWS Schema Conversion Tool, proyek konversi aplikasi adalah anak dari proyek konversi skema database. Setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak.

Note

AWS SCTtidak mendukung konversi antara sumber dan target berikut:

- Oracle para Oracle
- PostgreSQL ke PostgreSQL atau Aurora PostgreSQL
- MySQL ke MySQL

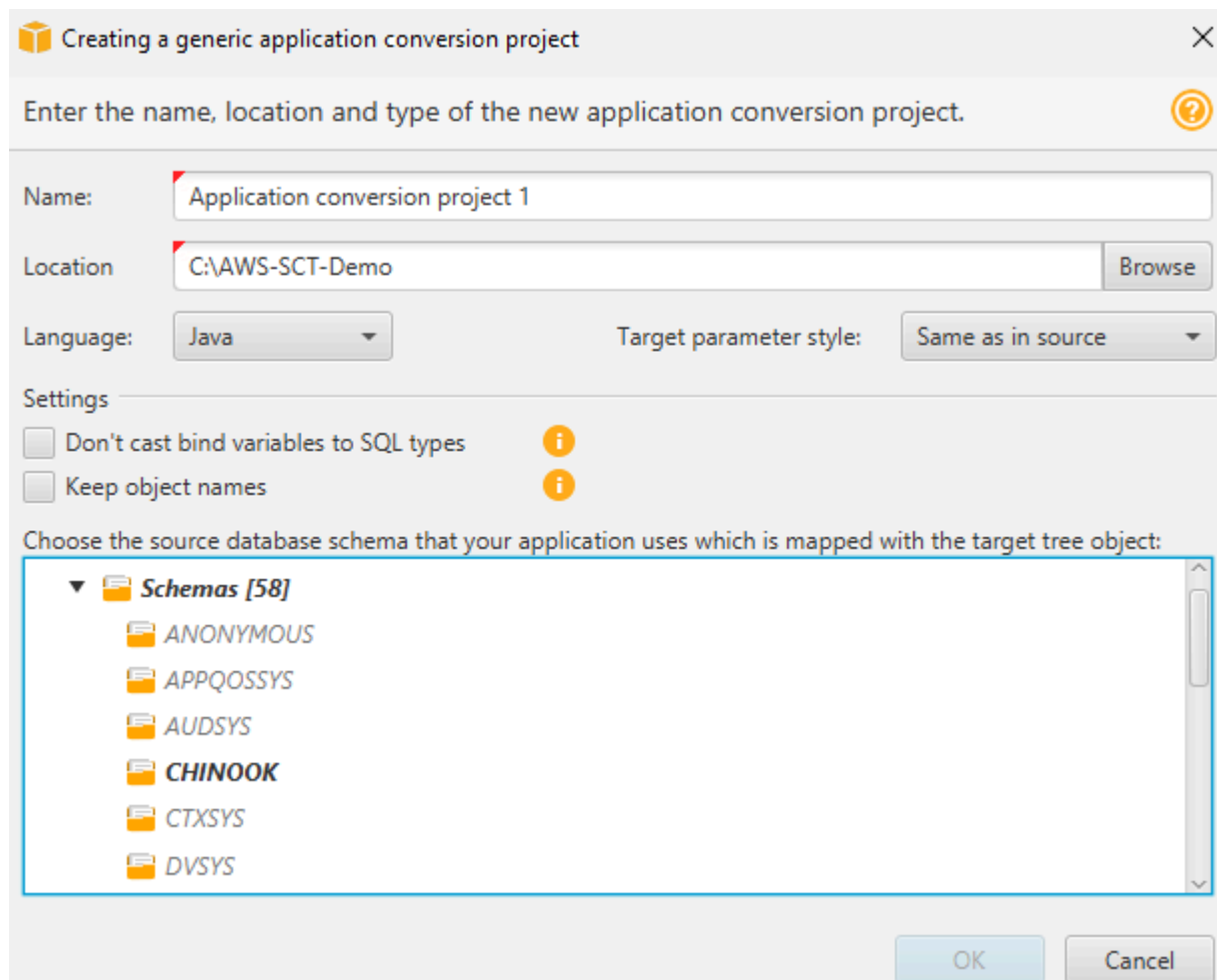
- SQL Server ke SQL Server
- Pergeseran Merah Amazon ke Pergeseran Merah
- SQL Server para Babelfish
- Layanan Integrasi SQL ServerAWS Glue
- Apache Cassandra ke Amazon DynamoDB

Gunakan prosedur berikut untuk membuat proyek konversi aplikasi generik.

Untuk membuat proyek konversi aplikasi

1. DalamAWS Schema Conversion Tool, pilihAplikasi generik baru padaAplikasimenu

TheProyek konversi aplikasi barukotak dialog muncul.



2. Menambahkan informasi proyek berikut.

Untuk parameter ini	Lakukan hal berikut
Nama	Masukkan nama untuk proyek konversi aplikasi. Setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak, jadi pilih nama yang masuk akal jika Anda menambahkan lebih banyak proyek nanti.
Lokasi	Masukkan lokasi kode sumber yang akan dibangun.
Bahasa	Pilih salah satu dari berikut: <ul style="list-style-type: none">• Java• C++• C#• Apa saja
Gaya parameter target	Pilih sintaks yang akan digunakan untuk mengikat variabel dalam kode yang dikonversi. Platform database yang berbeda menggunakan sintaks yang berbeda untuk variabel bind. Pilih salah satu opsi berikut: <ul style="list-style-type: none">• Sama seperti di sumber• Posisi (?)• Diindeks (:1)• Diindeks (\$1)• Dinamakan (@name)• Bernama (:nama)• Bernama (&nama)• Bernama (\$ nama)• Dinamakan (#name)• Dinamakan (! nama!)
Pilih skema basis data sumber	Di pohon sumber, pilih skema yang digunakan aplikasi Anda. Pastikan skema ini adalah bagian dari aturan pemetaan.

- Pilih **Jangan mentransmisikan variabel pengikat ke tipe SQL** untuk menghindari konversi jenis variabel bind ke tipe SQL. Opsi ini hanya tersedia untuk konversi Oracle ke PostgreSQL.

Misalnya, kode aplikasi sumber Anda menyertakan kueri Oracle berikut:

```
SELECT * FROM ACCOUNT WHERE id = ?
```

Saat Anda memilih **Jangan mentransmisikan variabel pengikat ke tipe SQL**, AWS SCT mengkonversi kueri ini seperti yang ditunjukkan.

```
SELECT * FROM account WHERE id = ?
```

Saat Anda membersihkan **Jangan mentransmisikan variabel pengikat ke tipe SQL**, AWS SCT mengubah tipe variabel bind ke **NUMERIC** tipe data. Hasil konversi ditampilkan sebagai berikut.

```
SELECT * FROM account WHERE id = (?)::NUMERIC
```

- Pilih **Simpan nama objek** untuk menghindari menambahkan nama skema ke nama objek yang dikonversi. Opsi ini hanya tersedia untuk konversi Oracle ke PostgreSQL.

Misalnya, misalkan kode aplikasi sumber Anda termasuk query Oracle berikut.

```
SELECT * FROM ACCOUNT
```

Saat Anda memilih **Simpan nama objek**, AWS SCT mengkonversi kueri ini seperti yang ditunjukkan.

```
SELECT * FROM account
```

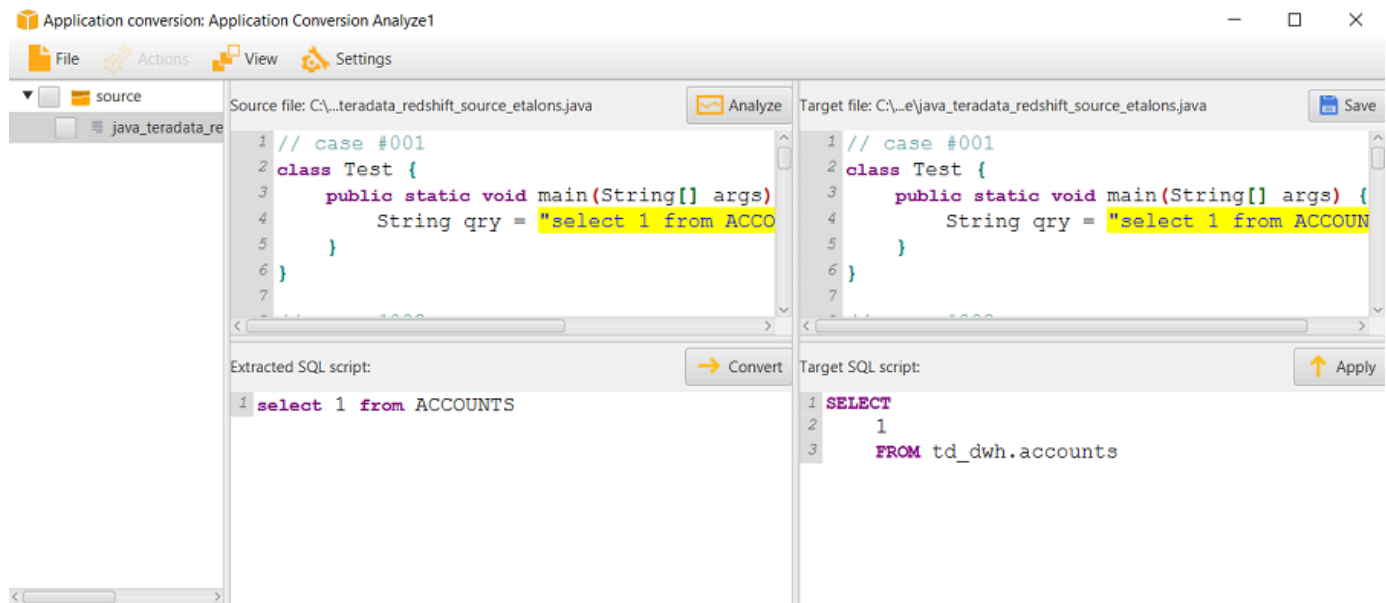
Saat Anda membersihkan **Simpan nama objek**, AWS SCT menambahkan nama skema untuk nama tabel. Hasil konversi ditampilkan sebagai berikut.

```
SELECT * FROM schema_name.account
```

Jika kode sumber Anda menyertakan nama objek induk dalam nama objek, AWS SCT menggunakan format ini dalam kode yang dikonversi. Dalam hal ini, abaikan **Simpan nama objek** pilihan karena AWS SCT menambahkan nama objek induk dalam kode yang dikonversi.

5. Pilih OKE untuk membuat proyek konversi aplikasi Anda.

Jendela proyek akan terbuka.



Mengelola proyek konversi aplikasi di AWS SCT

Anda dapat membuka proyek konversi aplikasi yang ada dan menambahkan beberapa proyek konversi aplikasi.

Setelah Anda membuat proyek konversi aplikasi, jendela proyek terbuka secara otomatis. Anda dapat menutup jendela proyek konversi aplikasi dan kembali lagi nanti.

Untuk membuka proyek konversi aplikasi yang ada

1. Di panel kiri, pilih menu konteks (klik kanan) dan buka menu konteks (klik kanan).
2. Pilih Kelola aplikasi.

Untuk menambahkan proyek konversi aplikasi tambahan

1. Di panel kiri, pilih menu konteks (klik kanan) dan buka menu konteks (klik kanan).
2. Pilih Aplikasi baru.
3. Masukkan informasi yang diperlukan untuk membuat proyek konversi aplikasi baru. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi generik](#).

Menganalisis dan mengonversi kode SQL Anda di AWS SCT

Gunakan prosedur berikut untuk menganalisis dan mengonversi kode SQL Anda di AWS Schema Conversion Tool.

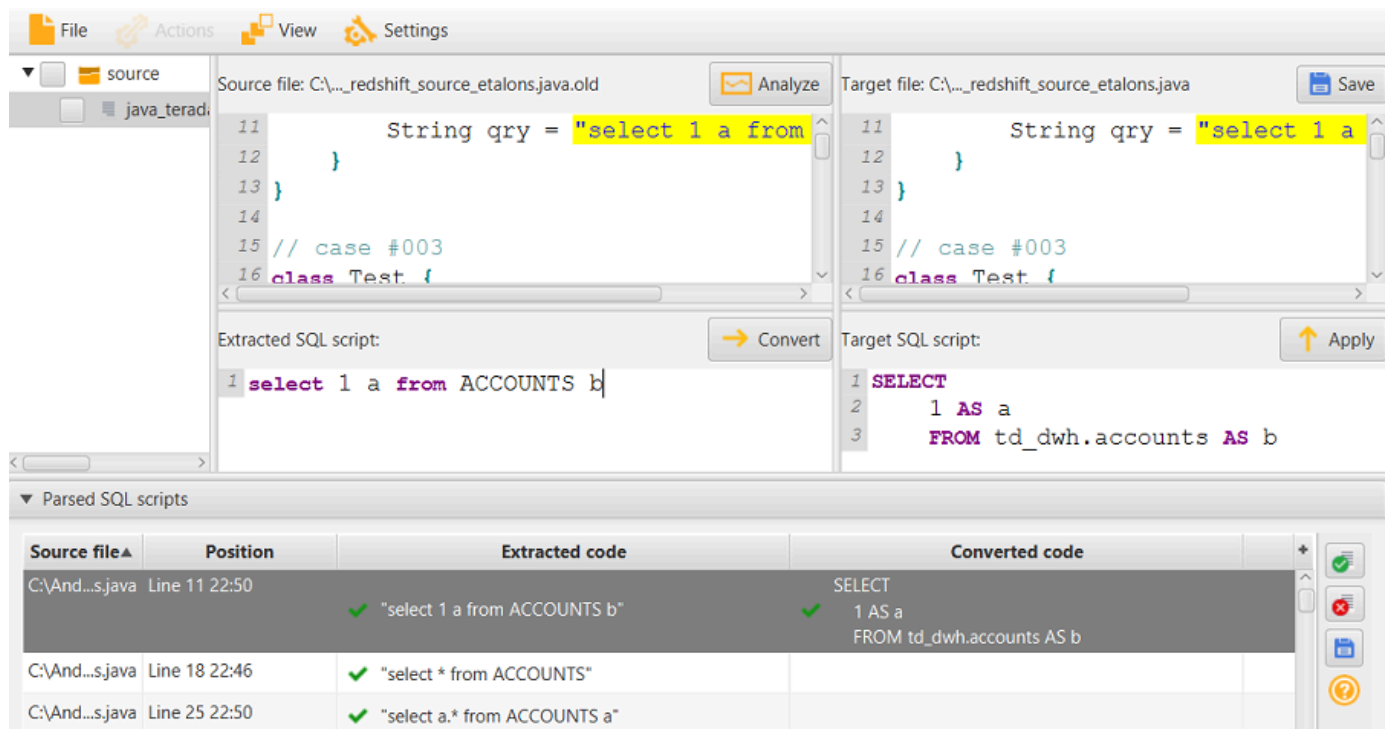
Untuk menganalisis dan mengonversi kode SQL Anda

1. Buka proyek konversi aplikasi yang ada, dan pilih **Menganalisis**.

AWS SCT menganalisis kode aplikasi Anda dan mengekstrak kode SQL. AWS SCT menampilkan kode SQL yang diekstraksi di Skrip SQL yang diuraikan.

2. Untuk Skrip SQL yang diurai, pilih item untuk meninjau kode SQL yang diekstraksi. AWS SCT menampilkan kode item yang dipilih di Skrip SQL yang diekstrak panel.
3. Pilih **Mengonversi** untuk mengonversi kode SQL. Skrip SQL yang diekstrak panel AWS SCT mengonversi kode ke format yang kompatibel dengan database target Anda.

Anda dapat mengedit kode SQL yang dikonversi. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan kode SQL Anda yang telah dikonversi](#).



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"	

4. Saat Anda membuat laporan penilaian konversi aplikasi, AWS SCT mengonversi semua item kode SQL diekstraksi. Untuk informasi selengkapnya, lihat [Membuat dan menggunakan laporan penilaian](#).

Membuat dan menggunakan AWS SCT laporan penilaian di AWS SCT

The laporan penilaian konversi aplikasi memberikan informasi tentang mengonversi kode SQL aplikasi ke format yang kompatibel dengan database target Anda. Laporan ini merinci semua kode SQL yang diekstraksi, semua kode SQL yang dikonversi, dan item tindakan untuk kode SQL yang AWS SCT tidak bisa mengkonversi.

Membuat laporan penilaian konversi aplikasi

Gunakan prosedur berikut untuk membuat laporan penilaian konversi aplikasi.

Untuk membuat laporan penilaian konversi aplikasi

1. Di jendela proyek konversi aplikasi, pilih **Buat laporan** pada **Tindakan** menu

AWS SCT membuat laporan penilaian konversi aplikasi dan membukanya di jendela proyek konversi aplikasi.

2. Tinjau **Ringkasan** Tinjau

The **Ringkasan** tab, ditampilkan berikut, menampilkan informasi ringkasan dari laporan penilaian aplikasi. Ini menunjukkan item kode SQL yang dikonversi secara otomatis, dan item yang tidak dikonversi secara otomatis.



3. Pilih **Tindakan ekstraksi SQL**.

Tinjau daftar kode kode yang AWS SCT tidak dapat mengekstrak dari kode sumber Anda.

4. Pilih **Tindakan konversi SQL**.

Tinjau daftar kode kode yang AWS SCT tidak dapat mengonversi secara otomatis. Gunakan tindakan yang disarankan untuk mengonversi kode SQL secara manual. Untuk informasi tentang cara mengedit kode SQL yang telah dikonversi, lihat [Mengedit dan menyimpan kode SQL Anda yang dikonversi dengan AWS SCT](#).

5. (Opsional) Simpan salinan lokal sebagai file PDF atau file nilai yang dipisahkan koma (CSV):
 - Pilih **Simpan ke PDF** di kanan atas untuk menyimpan laporan.

File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi aplikasi.

- Pilih **Simpan ke CSV** di kanan atas untuk menyimpan laporan.

File CSV berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi kode SQL.

Mengedit dan menyimpan kode SQL Anda yang dikonversi dengan AWS SCT

Laporan penilaian mencakup daftar kode yang tidak bisa dikonversi oleh AWS SCT. Untuk setiap item, AWS SCT membuat item tindakan pada Tindakan konversi SQL. Untuk item ini, Anda dapat mengedit kode SQL secara manual untuk melakukan konversi.

Gunakan prosedur berikut untuk mengedit kode SQL Anda yang dikonversi, menerapkan perubahan, dan kemudian menyimpannya.

Untuk mengedit, menerapkan perubahan ke, dan menyimpan kode SQL Anda yang telah dikonversi

1. Edit kode SQL Anda yang dikonversi langsung di Target skrip SQL panel. Jika tidak ada kode yang dikonversi yang ditampilkan, Anda dapat mengklik panel dan mulai mengetik.
2. Setelah Anda selesai mengedit kode SQL yang dikonversi, pilih **Terapkan**. Pada titik ini, perubahan disimpan dalam memori, tetapi belum ditulis ke file Anda.
3. Pilih **Simpan** untuk menyimpan perubahan Anda.

Ketika Anda memilih **Simpan**, Anda menerima file asli Anda. Buat salinan file asli Anda sebelum menyimpan sehingga Anda memiliki catatan kode aplikasi asli Anda.

Mengonversi kode SQL dalam aplikasi C # dengan AWS SCT

Untuk konversi Oracle ke PostgreSQL, Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi kode SQL yang disematkan ke dalam aplikasi C # Anda. Konverter aplikasi C # khusus ini memahami logika aplikasi. Ini mengumpulkan pernyataan yang terletak di objek aplikasi yang berbeda, seperti fungsi, parameter, variabel lokal, dan sebagainya.

Karena analisis mendalam ini, konverter kode SQL aplikasi C # memberikan hasil konversi yang lebih baik daripada konverter generik.

Membuat proyek konversi aplikasi C # diAWS SCT

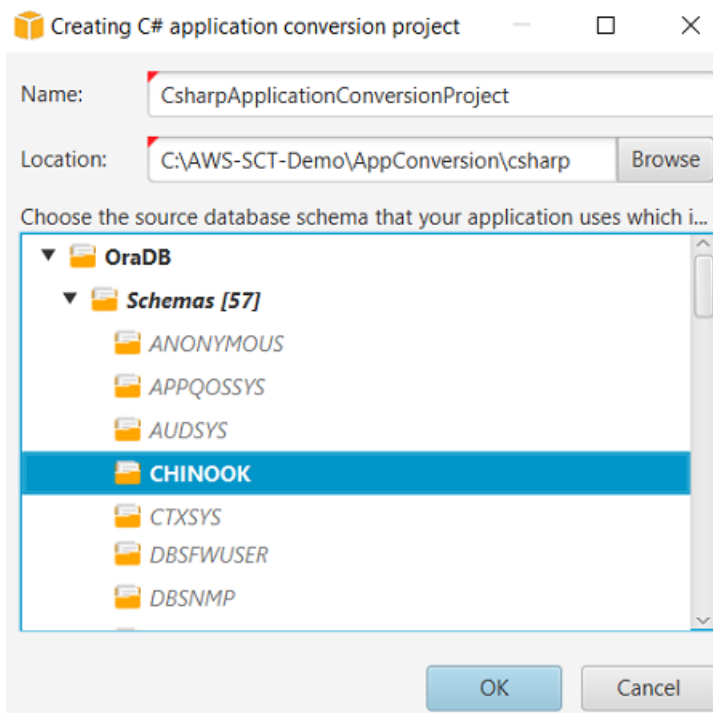
Anda dapat membuat proyek konversi aplikasi C # hanya untuk mengonversi skema database Oracle ke skema database PostgreSQL. Pastikan Anda menambahkan aturan pemetaan dalam proyek Anda yang menyertakan skema Oracle sumber dan database PostgreSQL target. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#).

Anda dapat menambahkan beberapa proyek konversi aplikasi dalam satuAWS SCTproyek Gunakan prosedur berikut untuk membuat proyek konversi aplikasi C #.

Untuk membuat proyek konversi aplikasi C #

1. Buat proyek konversi database, dan tambahkan database Oracle sumber. Untuk informasi selengkapnya, lihat [MembuatAWS SCT proyek](#) dan [Menambahkan server database keAWS SCT proyek](#).
2. Tambahkan aturan pemetaan yang mencakup database Oracle sumber Anda dan database PostgreSQL target. Anda dapat menambahkan database PostgreSQL target atau menggunakan platform database target PostgreSQL virtual dalam aturan pemetaan. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#) dan [Menggunakan target virtual](#).
3. PadaLihatmenu, pilihTampilan utama.
4. PadaAplikasimenu, pilihAplikasi C# baru.

TheMembuat proyek konversi aplikasi C #kotak dialog muncul.



5. Untuk Nama, masukkan nama untuk proyek konversi aplikasi. Karena setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak, pilih nama yang masuk akal jika Anda menambahkan beberapa proyek.
6. Untuk Lokasi, masukkan lokasi kode sumber yang akan dibangun.
7. Di pohon sumber, pilih skema yang digunakan aplikasi Anda. Pastikan bahwa skema ini adalah bagian dari aturan pemetaan. AWS SCT menyoroti skema yang merupakan bagian dari aturan pemetaan dalam huruf tebal.
8. Pilih OK untuk membuat proyek konversi aplikasi C # Anda.
9. Temukan proyek konversi aplikasi C# Anda di Aplikasi simpul di panel kiri.

Mengonversi kode SQL aplikasi C # Anda AWS SCT

Setelah Anda menambahkan aplikasi C # Anda ke AWS SCT proyek, mengonversi kode SQL dari aplikasi ini ke format yang kompatibel dengan platform database target Anda. Gunakan prosedur berikut untuk menganalisis dan mengonversi kode SQL yang disematkan dalam aplikasi C# Anda di AWS Schema Conversion Tool.

Untuk mengonversi kode SQL

1. Perluas C# simpul di bawah Aplikasi di panel kiri.

2. Pilih aplikasi untuk mengonversi, dan buka menu konteks (klik kanan).
3. Pilih **Mengkonversi**. AWS SCT menganalisis file kode sumber Anda, menentukan logika aplikasi, dan memuat metadata kode ke dalam proyek. Metadata kode ini mencakup kelas C #, objek, metode, variabel global, antarmuka, dan sebagainya.

Di panel basis data yang akan AWS SCT membuat struktur folder yang mirip dengan proyek aplikasi sumber Anda. Di sini Anda dapat meninjau kode aplikasi yang dikonversi.

```

Source Oracle file: SpecialEscapeSequences.cs
Properties Text Related converted objects Statistics Settings
18      {
19          string str1 = "SELECT *\n" +
20                      "FROM\t JAVADB.DATETYPE_MIXED_ALL\n\r" +
21                      "WHERE COL_CHAR = \ 'CHAR\ ' ";
22
23          command.CommandText = str1;
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 str1 = "SELECT *\n" +
21                      "FROM\t JAVADB.DATETYPE_MIXED_ALL\n\r" +
22                      "WHERE COL_CHAR = \ 'CHAR\ ' ";
23
24          command.CommandText = str1;
25          command.ExecuteNonQuery();
26      }
27      connection.Close();

```

4. Simpan kode aplikasi yang telah dikonversi. Untuk informasi selengkapnya, lihat [Menyimpan kode aplikasi Anda yang telah dikonversi](#).

Aplikasi C# Anda mungkin menyertakan kode SQL yang berinteraksi dengan database sumber yang berbeda. Anda dapat bermigrasi ke PostgreSQL beberapa database sumber ini. Dalam hal ini, pastikan Anda tidak mengonversi kode SQL yang berinteraksi dengan database yang Anda kecualikan dari lingkup migrasi. Anda dapat mengecualikan file sumber aplikasi C# Anda dari lingkup konversi. Untuk melakukannya, kosongkan kotak centang yang ingin Anda kecualikan dari lingkup konversi.

Setelah Anda mengubah cakupan konversi, AWS SCT masih menganalisis kode SQL semua file sumber aplikasi C # Anda. Kemudian, AWS SCT menyalin ke folder target semua file sumber yang Anda kecualikan dari lingkup konversi. Operasi ini memungkinkan untuk membangun aplikasi Anda setelah Anda menyimpan file aplikasi yang dikonversi.

Menyimpan kode aplikasi Anda yang dikonversi dengan AWS SCT

Gunakan prosedur berikut untuk menyimpan kode aplikasi Anda yang telah dikonversi.

Untuk menyimpan kode aplikasi yang telah dikonversi

1. Perluas **C#** simpul di bawah **Aplikasi** di panel database yang akan dibangun.
2. Pilih aplikasi Anda yang dikonversi, dan pilih **Simpan**.
3. Masukkan path ke folder untuk menyimpan kode aplikasi yang dikonversi, dan pilih **Pilih folder**.

Mengelola proyek konversi aplikasi C # di AWS SCT

Anda dapat menambahkan beberapa proyek konversi aplikasi C #, memperbarui kode aplikasi di AWS SCT proyek, atau hapus proyek konversi C# dari proyek AWS SCT proyek

Untuk menambahkan proyek konversi aplikasi C# tambahan

1. Perluas **Aplikasi** simpul di panel kiri.
2. Pilih **C#** simpul, dan buka menu konteks.
3. Pilih **Aplikasi baru**.
4. Masukkan informasi yang diperlukan untuk membuat proyek konversi aplikasi C # baru. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi C #](#).

Setelah Anda membuat perubahan dalam kode aplikasi sumber Anda, unggah ke AWS SCT proyek

Untuk mengunggah kode aplikasi yang akan diperbarui

1. Perluas **C#** simpul di bawah **Aplikasi** di panel kiri.
2. Pilih aplikasi untuk diperbarui, dan buka menu konteks (klik kanan).
3. Pilih **Segarkan** dan kemudian pilihnya.

AWS SCT mengunggah kode aplikasi Anda dari file sumber dan menghapus hasil konversi.

Untuk menyimpan perubahan kode yang Anda buat di AWS SCT dan hasil konversi, buat proyek konversi C# baru.

Untuk menghapus proyek konversi aplikasi C

1. PerluasC#simpul di bawahAplikasidi panel kiri.
2. Pilih aplikasi untuk menghapus, dan buka menu konteks (klik kanan).
3. PilihHapusdan kemudian memilihOKE.

Membuat laporan penilaian konversi aplikasi C# diAWS SCT

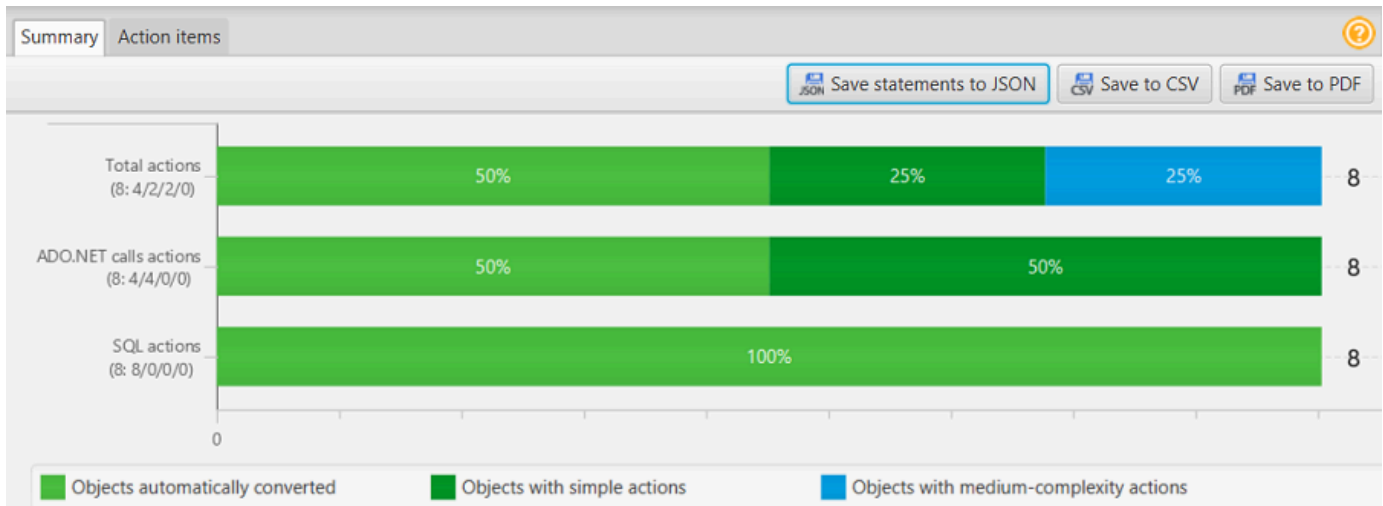
TheLaporan penilaian konversi aplikasi C #memberikan informasi tentang mengonversi kode SQL yang disematkan dalam aplikasi C# Anda ke format yang kompatibel dengan basis data target Anda. Laporan penilaian memberikan rincian konversi untuk semua titik eksekusi SQL dan semua file kode sumber. Laporan penilaian juga mencakup item tindakan untuk kode SQL yangAWS SCTtidak bisa mengkonversi.

Gunakan prosedur berikut untuk membuat laporan penilaian konversi aplikasi C #.

Untuk membuat laporan penilaian konversi aplikasi C#

1. PerluasC#simpul di bawahAplikasidi panel kiri.
2. Pilih aplikasi untuk mengonversi, dan buka menu konteks (klik kanan).
3. PilihMengkonversi.
4. PadaLihatmenu, pilihTampilan laporan penilaian.
5. TinjauRingkasanTinjau

TheRingkasantab, ditampilkan berikut, menampilkan informasi ringkasan eksekutif dari laporan penilaian aplikasi C#. Ini menunjukkan hasil konversi untuk semua titik eksekusi SQL dan semua file kode sumber.



6. Pilih **Simpan pernyataan ke JSON** untuk menyimpan kode SQL yang diekstrak dari aplikasi C# Anda sebagai file JSON.
7. (Opsional) Simpan salinan lokal sebagai file PDF atau file nilai yang dipisahkan koma (CSV):
 - Pilih **Simpan ke PDF** di kanan atas untuk menyimpan laporan.

File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi aplikasi.

- Pilih **Simpan ke CSV** di kanan atas untuk menyimpan laporan.

File CSV berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi kode SQL.

Mengonversi kode SQL dalam aplikasi C ++ dengan AWS SCT

Untuk konversi Oracle ke PostgreSQL, Anda dapat menggunakan AWS SCT untuk mengonversi kode SQL yang disematkan ke dalam aplikasi C++ Anda. Konverter aplikasi C++ khusus ini memahami logika aplikasi. Ini mengumpulkan pernyataan yang terletak di objek aplikasi yang berbeda, seperti fungsi, parameter, variabel lokal, dan sebagainya.

Karena analisis mendalam ini, konverter kode SQL aplikasi C++ memberikan hasil konversi yang lebih baik daripada konverter generik.

Membuat proyek konversi aplikasi C ++ di AWS SCT

Anda dapat membuat proyek konversi aplikasi C++ hanya untuk mengonversi skema database Oracle ke skema database PostgreSQL. Pastikan Anda menambahkan aturan pemetaan dalam

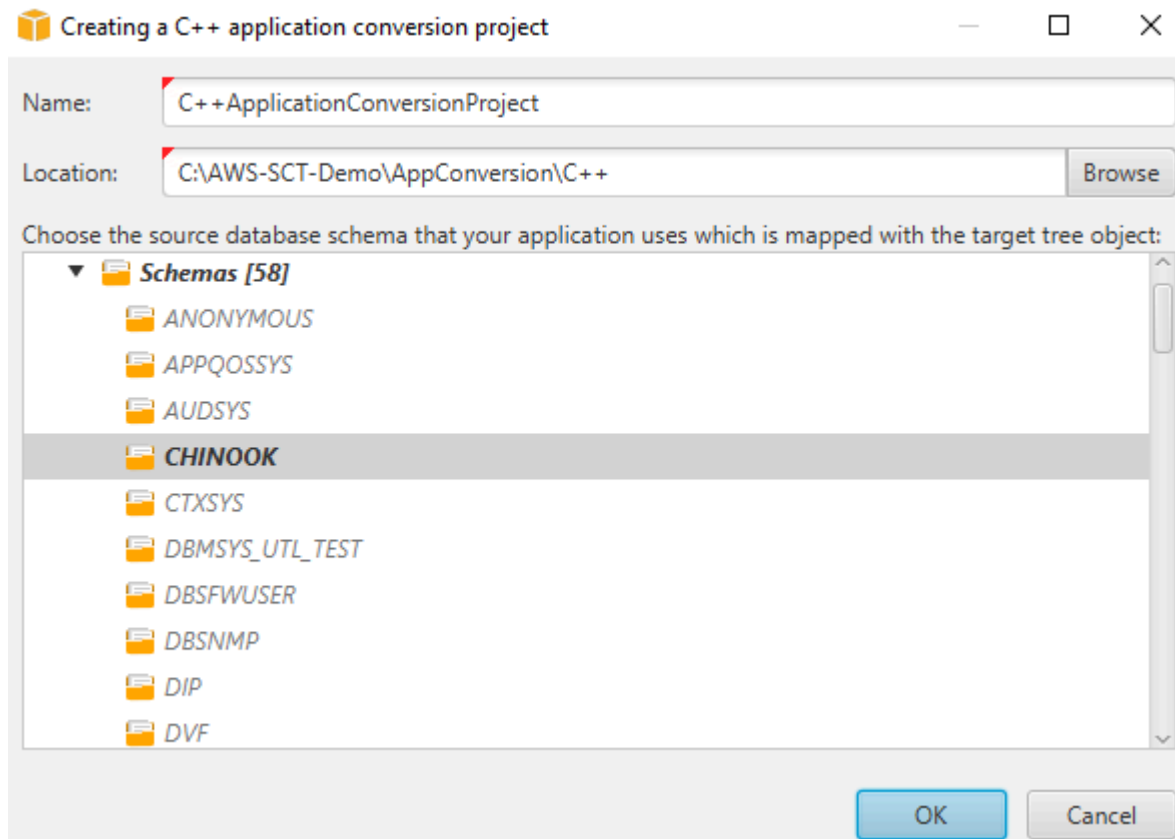
proyek Anda yang menyertakan skema Oracle sumber dan database PostgreSQL target. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#).

Anda dapat menambahkan beberapa proyek konversi aplikasi dalam satu AWS SCT proyek

Untuk membuat proyek konversi aplikasi C++

1. Buat proyek konversi database, dan tambahkan database Oracle sumber. Untuk informasi selengkapnya, lihat [Membuat AWS SCT proyek](#) dan [Menambahkan server database ke AWS SCT proyek](#).
2. Tambahkan aturan pemetaan yang mencakup database Oracle sumber Anda dan database PostgreSQL target. Anda dapat menambahkan database PostgreSQL target atau menggunakan platform database target PostgreSQL virtual dalam aturan pemetaan. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#) dan [Menggunakan target virtual](#).
3. Pada Lihat menu, pilih Tampilan utama.
4. Pada Aplikasi menu, pilih Aplikasi C++ baru.

Untuk membuat proyek konversi aplikasi C++ kotak dialog muncul.



5. Untuk **Nama**, masukkan nama untuk proyek konversi aplikasi. Karena setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak, pilih nama yang masuk akal jika Anda menambahkan beberapa proyek.
6. Untuk **Lokasi**, masukkan lokasi kode sumber untuk aplikasi Anda.
7. Di pohon sumber, pilih skema yang digunakan aplikasi Anda. Pastikan bahwa skema ini adalah bagian dari aturan pemetaan. AWS SCT menyoroti skema yang merupakan bagian dari aturan pemetaan dalam huruf tebal.
8. Pilih **OKE** untuk membuat proyek konversi aplikasi C++ Anda.
9. Temukan proyek konversi aplikasi C++ Anda di **Aplikasi** simpul di panel kiri.

Mengonversi kode SQL aplikasi C++ Anda AWS SCT

Setelah Anda menambahkan aplikasi C++ Anda ke AWS SCT proyek, mengkonversi kode SQL dari aplikasi ini ke format yang kompatibel dengan platform database target Anda. Gunakan prosedur berikut untuk menganalisis dan mengonversi kode SQL yang disematkan dalam aplikasi C++ Anda di AWS SCT.

Untuk mengonversi kode SQL

1. Perluas **C++** simpul di bawah **Aplikasi** di panel kiri, dan pilih aplikasi untuk dikonversi.
2. Dalam **Sumber** proyek aplikasi Oracle, pilih **Pengaturan**. Tinjau dan edit pengaturan konversi untuk aplikasi C++ yang dipilih. Anda juga dapat menentukan pengaturan konversi untuk semua aplikasi C++ yang Anda tambahkan ke AWS SCT proyek. Untuk informasi selengkapnya, lihat [Mengelola proyek konversi aplikasi C++](#).
3. Untuk **Jenis** kompilasi, pilih kompilasi yang Anda gunakan untuk kode sumber aplikasi C++ Anda. AWS SCT mendukung kompilasi C++ berikut: Microsoft Visual C++, GCC, Koleksi Kompilasi GNU, dan **Dentang**. Opsi defaultnya adalah Microsoft Visual C++.
4. Untuk **Makro** yang ditentukan pengguna, masukkan path ke file yang menyertakan makro yang ditentukan pengguna dari proyek C++ Anda. Pastikan file ini memiliki struktur berikut: `#define name value`. Dalam contoh sebelumnya, `value` adalah parameter opsional. Nilai default untuk parameter opsional ini adalah `1`.

Untuk membuat file ini, buka proyek Anda di Microsoft Visual Studio, lalu pilih **Proyek**, **Properti**, **C/C++**, dan **Preprosesor**. Untuk **Definisi preprosesor**, pilih **Sunting** dan salin nama dan nilai ke file teks baru. Kemudian, untuk setiap string dalam file, tambahkan awalan berikut: `#define` .

- Untuk Eksternal termasuk direktori, masukkan jalur ke folder yang menyertakan pustaka eksternal yang Anda gunakan dalam proyek C++ Anda.
- Di panel kiri, pilih menu konteks (klik kanan).
- Pilih **Mengkonversi AWS SCT** menganalisis file kode sumber Anda, menentukan logika aplikasi, dan memuat metadata kode ke dalam proyek. Metadata kode ini mencakup kelas C++, objek, metode, variabel global, antarmuka, dan sebagainya.

Di panel basis data yang akan AWS SCT membuat struktur folder yang mirip dengan proyek aplikasi sumber Anda. Di sini Anda dapat meninjau kode aplikasi yang dikonversi, seperti yang ditunjukkan berikut.

The screenshot displays two panels comparing the original C++ source code with its converted PostgreSQL equivalent. The top panel, titled 'Source Oracle file: StringInitialization.cpp', shows the original code with line numbers 44 through 55. The bottom panel, titled 'Target Amazon RDS for PostgreSQL file: StringInitialization.cpp', shows the converted code with line numbers 45 through 56. The converted code maintains the same logic but uses PostgreSQL-compatible syntax, such as 'javadb.get_int()' instead of 'GET_INT()'.

```

Source Oracle file: StringInitialization.cpp
44     if ((dRet == SQLDriverConnect(hDBc, NULL, lpConnectionStr, connectionStr.size(), OutConnStr, 0xFF
45     {
46         SQLHANDLE hSelectStm = NULL;
47
48         if ((dRet = SQLAllocHandle(SQL_HANDLE_STMT, hDBc, &hSelectStm)) == SQL_SUCCESS)
49         {
50
51             char* buff = static_cast<char*>(malloc(0xFF * sizeof(char)));
52             strncpy_s(&buff[0], 0xFF, "SELECT JAVADB.GET_INT() FROM DUAL", 18);
53
54             if ((dRet = SQLExecDirect(hSelectStm, buff, strlen(buff))) == SQL_SUCCESS)
55             {
56
Target Amazon RDS for PostgreSQL file: StringInitialization.cpp
45     if ((dRet == SQLDriverConnect(hDBc, NULL, lpConnectionStr, connectionStr.size(), OutConnStr,
46     {
47         SQLHANDLE hSelectStm = NULL;
48
49         if ((dRet = SQLAllocHandle(SQL_HANDLE_STMT, hDBc, &hSelectStm)) == SQL_SUCCESS)
50         {
51
52             char* buff = static_cast<char*>(malloc(0xFF * sizeof(char)));
53             strncpy_s(&buff[0], 0xFF, "SELECT javadb.get_int()", 18);
54
55             if ((dRet = SQLExecDirect(hSelectStm, buff, strlen(buff))) == SQL_SUCCESS)
56             {

```

- Simpan kode aplikasi yang telah dikonversi. Untuk informasi selengkapnya, lihat [Menyimpan kode aplikasi Anda yang telah dikonversi](#).

Menyimpan kode aplikasi Anda yang dikonversi dengan AWS SCT

Gunakan prosedur berikut untuk menyimpan kode aplikasi Anda yang telah dikonversi.

Untuk menyimpan kode aplikasi yang telah dikonversi

1. Perluas C++ simpul di bawah Aplikasi di panel database yang akan dibangun.
2. Pilih aplikasi Anda yang dikonversi, dan pilih Simpan.
3. Masukkan path ke folder untuk menyimpan kode aplikasi yang dikonversi, dan pilih Pilih folder.

Mengelola proyek konversi aplikasi C ++ di AWS SCT

Anda dapat menambahkan beberapa proyek konversi aplikasi C++, mengedit pengaturan konversi, memperbarui kode aplikasi C ++, atau menghapus proyek konversi C ++ dari AWS SCT proyek

Untuk menambahkan proyek konversi aplikasi C++ tambahan

1. Perluas Aplikasi simpul di panel kiri.
2. Pilih C++ simpul, dan buka menu konteks.
3. Pilih Aplikasi baru.
4. Masukkan informasi yang diperlukan untuk membuat proyek konversi aplikasi C ++ baru. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi C ++](#).

Anda dapat menentukan pengaturan konversi untuk semua proyek konversi aplikasi C ++ di AWS SCT proyek

Untuk mengedit pengaturan konversi untuk semua aplikasi C++

1. Pada Pengaturan menu, pilih Pengaturan proyek, dan kemudian pilih Konversi aplikasi.
2. Untuk Jenis kompiler, pilih kompiler yang Anda gunakan untuk kode sumber aplikasi C ++ Anda. AWS SCT mendukung kompiler C ++ berikut: Microsoft Visual C++, GCC, Koleksi Kompiler GNU, dan Dentang. Opsi defaultnya adalah Microsoft Visual C++.
3. Untuk Makro yang ditentukan pengguna, masukkan path ke file yang menyertakan makro yang ditentukan pengguna dari proyek C++ Anda. Pastikan file ini memiliki struktur berikut: `#define name value`. Dalam contoh sebelumnya, `value` adalah parameter opsional. Nilai default untuk parameter opsional ini adalah `1`.

Untuk membuat file ini, buka proyek Anda di Microsoft Visual Studio, lalu pilih Proyek, Properti, C/C++, dan Preprosesor. Untuk Definisi preprosesor, pilih Sunting dan salin nama dan nilai ke file teks baru. Kemudian, untuk setiap string dalam file, tambahkan awalan berikut: `#define` .

4. Untuk Eksternal termasuk direktori, masukkan jalur ke folder yang menyertakan pustaka eksternal yang Anda gunakan dalam proyek C++ Anda.
5. Pilih OKE untuk menyimpan pengaturan proyek dan menutup jendela.

Atau Anda dapat menentukan pengaturan konversi untuk setiap proyek konversi aplikasi C++. Untuk informasi selengkapnya, lihat [Mengonversi kode SQL aplikasi C++ Anda](#).

Setelah Anda membuat perubahan dalam kode aplikasi sumber Anda, unggah ke AWS SCT proyek

Untuk mengunggah kode aplikasi yang akan diperbarui

1. Perluas C++ simpul di bawah Aplikasi di panel kiri.
2. Pilih aplikasi untuk diperbarui, dan buka menu konteks (klik kanan).
3. Pilih Segarkan dan kemudian pilihnya.

AWS SCT mengunggah kode aplikasi Anda dari file sumber dan menghapus hasil konversi. Untuk menyimpan perubahan kode yang Anda buat AWS SCT dan hasil konversi, buat proyek konversi C++ baru.

Juga AWS SCT menghapus pengaturan konversi aplikasi yang Anda tentukan untuk aplikasi yang dipilih. Setelah Anda mengunggah kode aplikasi yang diperbarui, AWS SCT menerapkan nilai default dari pengaturan proyek.

Untuk menghapus proyek konversi aplikasi C++

1. Perluas C++ simpul di bawah Aplikasi di panel kiri.
2. Pilih aplikasi untuk menghapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus dan kemudian pilih OKE.

Membuat laporan penilaian konversi aplikasi C++ di AWS SCT

The Laporan penilaian konversi aplikasi C++ memberikan informasi tentang mengonversi kode SQL yang disematkan dalam aplikasi C++ Anda ke format yang kompatibel dengan basis data target Anda. Laporan penilaian memberikan rincian konversi untuk semua titik eksekusi SQL dan semua file kode sumber. Laporan penilaian juga mencakup item tindakan untuk kode SQL yang AWS SCT tidak bisa mengkonversi.

Untuk membuat laporan penilaian konversi aplikasi C++

1. Perluas C++ simpul di bawah Aplikasi di panel kiri.
2. Pilih aplikasi untuk mengonversi, dan buka menu konteks (klik kanan).
3. Pilih Mengkonversi.
4. Pada Lihat menu, pilih Tampilan laporan penilaian.
5. Tinjau Ringkasan Tinjau

The Ringkas tab menampilkan informasi ringkasan eksekutif dari laporan penilaian aplikasi C+++. Ini menunjukkan hasil konversi untuk semua titik eksekusi SQL dan semua file kode sumber.

6. Pilih Simpan pernyataan ke JSON untuk menyimpan kode SQL yang diekstrak dari aplikasi Java Anda sebagai file JSON.
7. (Opsional) Simpan salinan lokal sebagai file PDF atau file nilai yang dipisahkan koma (CSV):
 - Pilih Simpan ke PDF di kanan atas untuk menyimpan laporan.

File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi aplikasi.

- Pilih Simpan ke CSV di kanan atas untuk menyimpan laporan.

File CSV berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi kode SQL.

Mengonversi kode SQL dalam aplikasi Java dengan AWS SCT

Untuk konversi Oracle ke PostgreSQL, Anda dapat menggunakan AWS Schema Conversion Tool untuk mengonversi kode SQL yang disematkan ke dalam aplikasi Java Anda. Konverter aplikasi Java khusus ini memahami logika aplikasi. Ini mengumpulkan pernyataan yang terletak di objek aplikasi yang berbeda, seperti fungsi, parameter, variabel lokal, dan sebagainya.

Karena analisis mendalam ini, aplikasi Java SQL code converter memberikan hasil konversi yang lebih baik dibandingkan dengan konverter generik.

Jika aplikasi Java Anda menggunakan MyBatis kerangka kerja untuk berinteraksi dengan database, maka Anda dapat menggunakan AWS SCT untuk mengonversi pernyataan SQL yang disematkan ke MyBatis File XHTML dan anotasi. Untuk memahami logika pernyataan SQL ini, AWS SCT menggunakan MyBatis file konfigurasi. AWS SCT dapat secara otomatis menemukan file ini di folder aplikasi Anda, atau Anda dapat memasukkan jalur ke file ini secara manual.

Membuat proyek konversi aplikasi Java diAWS SCT

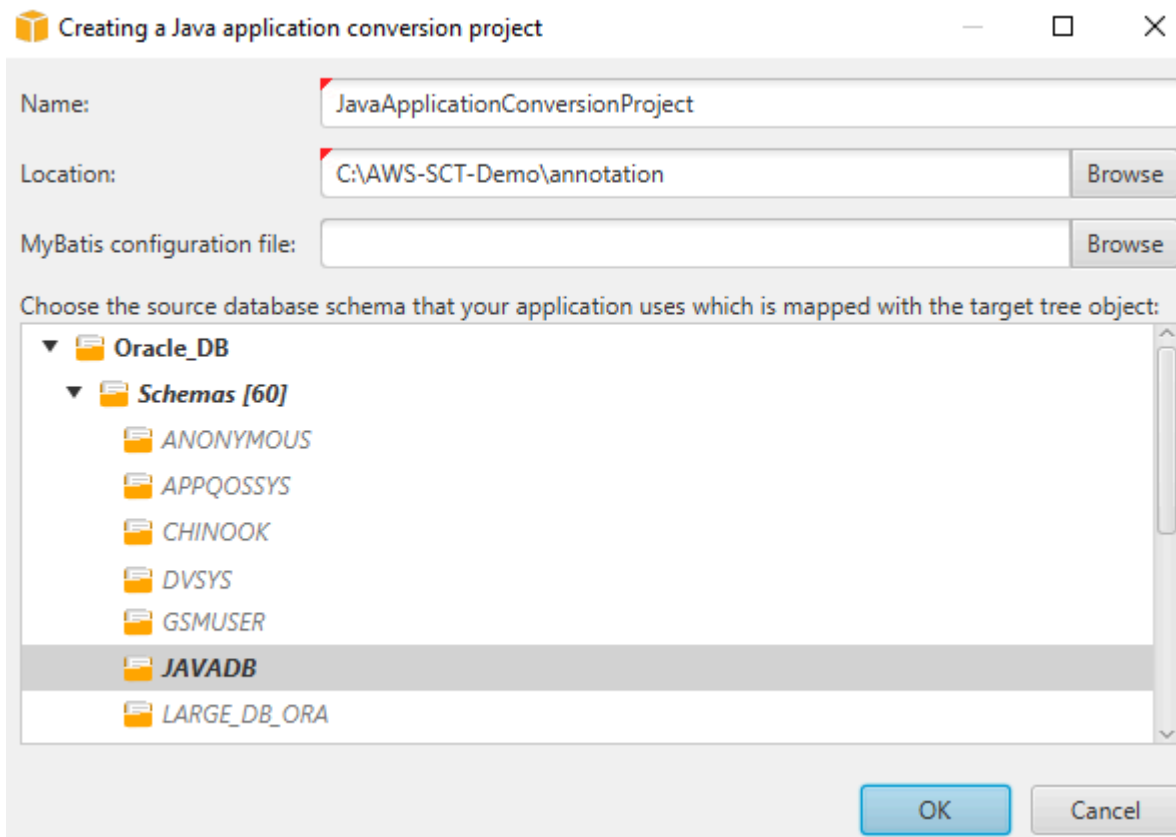
Anda dapat membuat proyek konversi aplikasi Java hanya untuk mengonversi skema database Oracle ke skema database PostgreSQL. Pastikan Anda menambahkan aturan pemetaan dalam proyek Anda yang menyertakan skema Oracle sumber dan database PostgreSQL target. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#).

Anda dapat menambahkan beberapa proyek konversi aplikasi dalam satuAWS SCTproyek. Gunakan prosedur berikut untuk membuat proyek konversi aplikasi Java.

Untuk membuat proyek konversi aplikasi Java

1. Buat proyek konversi database, dan tambahkan database Oracle sumber. Untuk informasi selengkapnya, lihat [MembuatAWS SCT proyek](#) dan [Menambahkan server database keAWS SCT proyek](#).
2. Tambahkan aturan pemetaan yang mencakup database Oracle sumber Anda dan database PostgreSQL target. Anda dapat menambahkan database PostgreSQL target atau menggunakan platform database target PostgreSQL virtual dalam aturan pemetaan. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#) dan [Menggunakan target virtual](#).
3. PadaLihatmenu, pilihTampilan utama.
4. PadaAplikasimenu, pilihAplikasi Java baru.

TheMembuat proyek konversi aplikasi Javakotak dialog muncul.



5. UntukNama, masukkan nama untuk proyek konversi aplikasi. Karena setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak, pilih nama yang masuk akal jika Anda menambahkan beberapa proyek.
6. UntukLokasi, masukkan lokasi kode sumber untuk aplikasi Anda.
7. (Opsional) UntukMyBatis berkas konfigurasi, masuki jalan menuju MyBatis file konfigurasi.AWS SCTmemindai folder aplikasi Anda untuk menemukan file ini secara otomatis. Jika file ini tidak terletak di folder aplikasi Anda, atau jika Anda menggunakan beberapa file konfigurasi, maka masukkan path secara manual.
8. Di pohon sumber, pilih skema yang digunakan aplikasi Anda. Pastikan bahwa skema ini adalah bagian dari aturan pemetaan.AWS SCTmenyoroti skema yang merupakan bagian dari aturan pemetaan dalam huruf tebal.
9. PilihOKEuntuk membuat proyek konversi aplikasi Java Anda.
10. Temukan proyek konversi aplikasi Java Anda diAplikasisimpul di panel kiri.

Mengonversi kode SQL aplikasi Java Anda AWS SCT

Setelah Anda menambahkan aplikasi Java Anda ke AWS SCT proyek, mengonversi kode SQL dari aplikasi ini ke format yang kompatibel dengan platform database target Anda. Gunakan prosedur berikut untuk menganalisis dan mengonversi kode SQL yang disematkan dalam aplikasi Java Anda di AWS Schema Conversion Tool.

Untuk mengonversi kode SQL

1. Perluas Java simpul di bawah Aplikasi di panel kiri.
2. Pilih aplikasi untuk mengonversi, dan buka menu konteks (klik kanan).
3. Pilih **Mengonversi AWS SCT** menganalisis file kode sumber Anda, menentukan logika aplikasi, dan memuat metadata kode ke dalam proyek. Metadata kode ini mencakup kelas Java, objek, metode, variabel global, antarmuka, dan sebagainya.

Di panel basis data yang akan AWS SCT membuat struktur folder yang mirip dengan proyek aplikasi sumber Anda. Di sini Anda dapat meninjau kode aplikasi yang dikonversi.

```
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. Simpan kode aplikasi yang telah dikonversi. Untuk informasi selengkapnya, lihat [Menyimpan kode aplikasi Anda yang telah dikonversi](#).

Aplikasi Java Anda mungkin menyertakan kode SQL yang berinteraksi dengan database sumber yang berbeda. Anda dapat bermigrasi ke PostgreSQL beberapa database sumber ini. Dalam hal ini, pastikan Anda tidak mengonversi kode SQL yang berinteraksi dengan database yang Anda keculikan dari lingkup migrasi. Anda dapat mengecualikan file sumber aplikasi Java Anda dari lingkup konversi. Untuk melakukannya, kosongkan kotak centang yang ingin Anda keculikan dari lingkup konversi.

Setelah Anda mengubah cakupan konversi, AWS SCT masih menganalisis kode SQL semua file sumber aplikasi Java Anda. Kemudian, AWS SCT menyalin ke folder target semua file sumber yang Anda keculikan dari lingkup konversi. Operasi ini memungkinkan untuk membangun aplikasi Anda setelah Anda menyimpan file aplikasi yang dikonversi.

Menyimpan kode aplikasi Anda yang dikonversi dengan AWS SCT

Gunakan prosedur berikut untuk menyimpan kode aplikasi Anda yang telah dikonversi.

Untuk menyimpan kode aplikasi yang telah dikonversi

1. Perluas **Javasimpul** di bawah **Aplikasi** di panel database yang akan dibangun.
2. Pilih aplikasi Anda yang dikonversi, dan pilih **Simpan**.
3. Masukkan path ke folder untuk menyimpan kode aplikasi yang dikonversi, dan pilih **Pilih folder**.

Jika aplikasi Java sumber Anda menggunakan MyBatis framework, pastikan bahwa Anda memperbarui file konfigurasi Anda untuk bekerja dengan database baru Anda.

Mengelola proyek konversi aplikasi Java di AWS SCT

Anda dapat menambahkan beberapa proyek konversi aplikasi Java, memperbarui kode aplikasi di AWS SCT, memproyeksikan, atau menghapus proyek konversi Java dari AWS SCT proyek.

Untuk menambahkan proyek konversi aplikasi Java tambahan

1. Perluas **Aplikasi** di panel kiri.
2. Pilih **Javasimpul**, dan buka menu konteks.
3. Pilih **Aplikasi baru**.
4. Masukkan informasi yang diperlukan untuk membuat proyek konversi aplikasi Java baru. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi Java](#).

Setelah Anda membuat perubahan dalam kode aplikasi sumber Anda, unggah ke AWS SCT proyek.

Untuk mengunggah kode aplikasi yang akan diperbarui

1. Perluas **Javasimpul** di bawah **Aplikasi** di panel kiri.
2. Pilih aplikasi untuk diperbarui, dan buka menu konteks (klik kanan).
3. Pilih **Segarkan** dan kemudian pilihnya.

AWS SCT mengunggah kode aplikasi Anda dari file sumber dan menghapus hasil konversi.

Untuk menyimpan perubahan kode yang Anda buat di AWS SCT dan hasil konversi, buat proyek konversi Java baru.

Jika aplikasi Java sumber Anda menggunakan MyBatis kerangka kerja,AWS SCTmenggunakan MyBatis file konfigurasi untuk mengurai kode SQL Anda. Setelah Anda mengubah file ini, unggah keAWS SCTproyek

Untuk mengedit jalur ke MyBatis berkas konfigurasi

1. PerluasJavasimpul di bawahAplikasidi panel kiri.
2. Pilih aplikasi Anda, lalu pilihPengaturan.
3. PilihJelajahi, dan kemudian pilih MyBatisfile konfigurasi.
4. Pilih Apply (Terapkan).
5. Di panel kiri, pilih menu konteks (klik kanan), dan pilih menu konteks (klik kanan), dan pilihSegarkan.

Untuk menghapus proyek konversi aplikasi Java

1. PerluasJavasimpul di bawahAplikasidi panel kiri.
2. Pilih aplikasi untuk menghapus, dan buka menu konteks (klik kanan).
3. PilihHapusdan kemudian memilihOKE.

Membuat laporan penilaian konversi aplikasi Java diAWS SCT

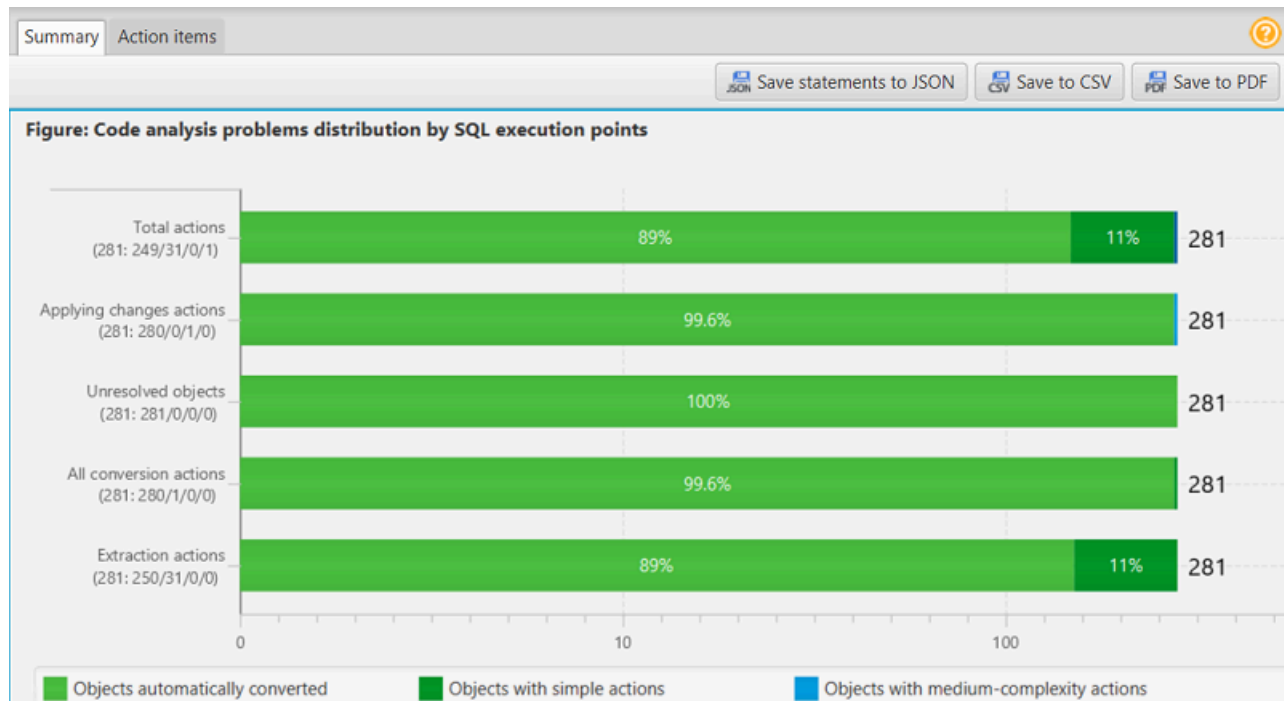
TheLaporan penilaian konversi aplikasi Javamemberikan informasi tentang mengonversi kode SQL yang disematkan dalam aplikasi Java Anda ke format yang kompatibel dengan basis data target Anda. Laporan penilaian memberikan rincian konversi untuk semua titik eksekusi SQL dan semua file kode sumber. Laporan penilaian juga mencakup item tindakan untuk kode SQL yangAWS SCTtidak bisa mengkonversi.

Gunakan prosedur berikut untuk membuat laporan penilaian konversi aplikasi Java.

Untuk membuat laporan penilaian konversi aplikasi

1. PerluasJavasimpul di bawahAplikasidi panel kiri.
2. Pilih aplikasi untuk mengonversi, dan buka menu konteks (klik kanan).
3. PilihMengkonversi.
4. PadaLihatmenu, pilihTampilan laporan penilaian.
5. TinjauRingkasanTinjau

TheRingkasantab, ditampilkan berikut, menampilkan informasi ringkasan eksekutif dari laporan penilaian aplikasi Java. Ini menunjukkan hasil konversi untuk semua titik eksekusi SQL dan semua file kode sumber.



6. Pilih Simpan pernyataan ke JSON untuk menyimpan kode SQL yang diekstrak dari aplikasi Java Anda sebagai file JSON.
7. (Opsional) Simpan salinan lokal sebagai file PDF atau file nilai yang dipisahkan koma (CSV):
 - Pilih Simpan ke PDF di kanan atas untuk menyimpan laporan.
File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi aplikasi.
 - Pilih Simpan ke CSV di kanan atas untuk menyimpan laporan.
File CSV berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi kode SQL.

Mengonversi kode SQL dalam aplikasi Pro* C dengan AWS SCT

Untuk konversi Oracle ke PostgreSQL, Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi kode SQL yang disematkan ke dalam aplikasi Pro* C Anda. Konverter aplikasi Pro* C khusus ini memahami logika aplikasi. Ini mengumpulkan pernyataan yang terletak di objek aplikasi yang berbeda, seperti fungsi, parameter, variabel lokal, dan sebagainya.

Karena analisis mendalam ini, konverter kode SQL aplikasi Pro* C memberikan hasil konversi yang lebih baik dibandingkan dengan konverter generik.

Membuat proyek konversi aplikasi Pro* C diAWS SCT

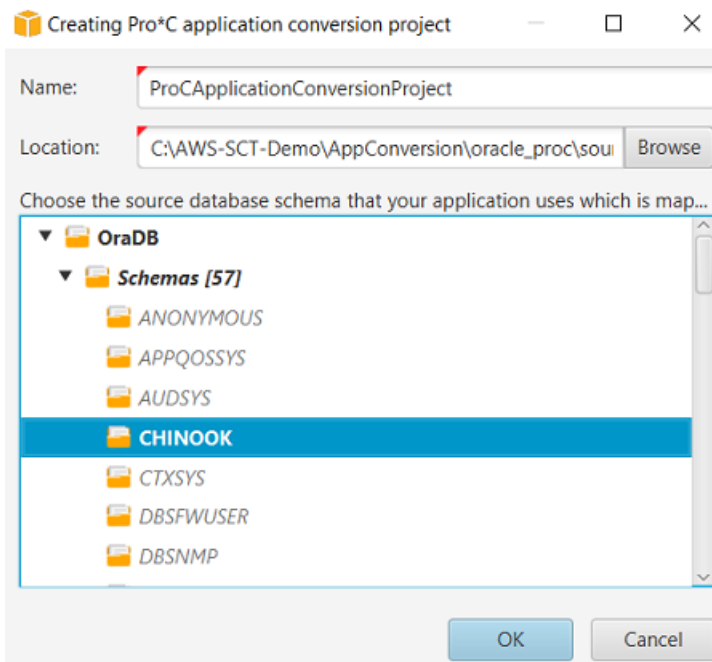
Anda dapat membuat proyek konversi aplikasi Pro* C hanya untuk mengonversi skema database Oracle ke skema database PostgreSQL. Pastikan Anda menambahkan aturan pemetaan dalam proyek Anda yang menyertakan skema Oracle sumber dan database PostgreSQL target. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#).

Anda dapat menambahkan beberapa proyek konversi aplikasi dalam satuAWS SCTproyek Gunakan prosedur berikut untuk membuat proyek konversi aplikasi Pro* C.

Untuk membuat proyek konversi aplikasi Pro*C

1. Buat proyek konversi database, dan tambahkan database Oracle sumber. Untuk informasi selengkapnya, lihat [MembuatAWS SCT proyek](#) dan [Menambahkan server database keAWS SCT proyek](#).
2. Tambahkan aturan pemetaan yang mencakup database Oracle sumber Anda dan database PostgreSQL target. Anda dapat menambahkan database PostgreSQL target atau menggunakan platform database target PostgreSQL virtual dalam aturan pemetaan. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan di AWS SCT](#) dan [Menggunakan target virtual](#).
3. PadaLihatmenu, pilihTampilan utama.
4. PadaAplikasimenu, pilihAplikasi Pro* C baru.

TheMembuat proyek konversi aplikasi Pro*Ckotak dialog muncul.



5. Untuk Nama, masukkan nama untuk proyek konversi aplikasi Pro* C. Karena setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak, pilih nama yang masuk akal jika Anda menambahkan beberapa proyek.
6. Untuk Lokasi, masukkan lokasi kode sumber untuk aplikasi Anda.
7. Di pohon sumber, pilih skema yang digunakan aplikasi Anda. Pastikan skema ini adalah bagian dari aturan pemetaan. AWS SCT menyoroti skema yang merupakan bagian dari aturan pemetaan dalam huruf tebal.
8. Pilih OK untuk membuat proyek konversi aplikasi Pro* C Anda.
9. Temukan proyek konversi aplikasi Pro* C Anda di Aplikasi simpul di panel kiri.

Mengonversi kode SQL aplikasi Pro* C Anda AWS SCT

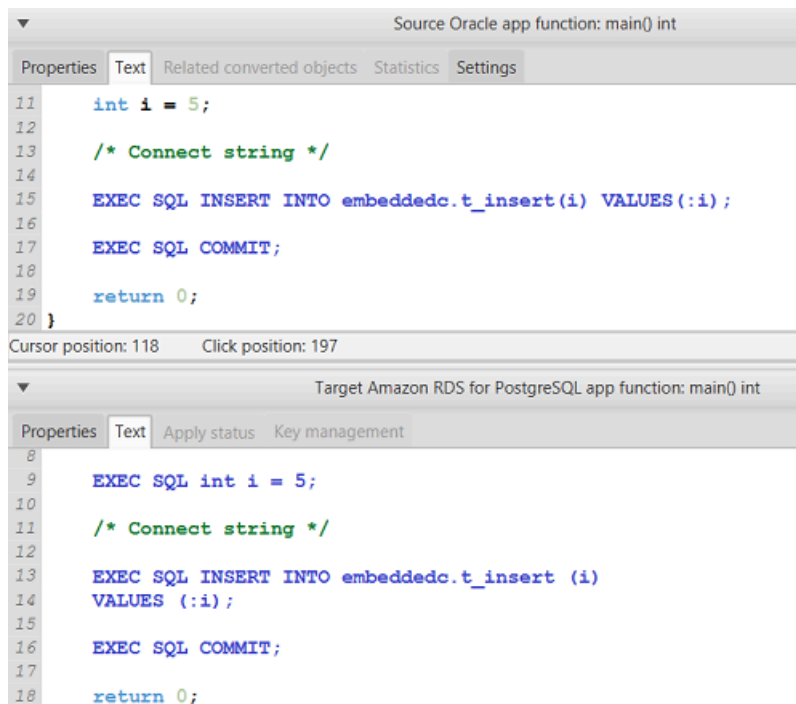
Setelah Anda menambahkan aplikasi Pro* C Anda ke AWS SCT proyek, mengkonversi kode SQL dari aplikasi ini ke format yang kompatibel dengan platform database target Anda. Gunakan prosedur berikut untuk menganalisis dan mengonversi kode SQL yang disematkan dalam aplikasi Pro* C Anda di AWS Schema Conversion Tool.

Untuk mengonversi kode SQL

1. Perluas Pro* C simpul di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan dikonversi dan kemudian pilih Pengaturan.

- a. Untuk Jalur file header global, masukkan path ke file header yang digunakan proyek aplikasi Anda.
 - b. Pilih Menafsirkan semua variabel host yang belum terselesaikan sebagai untuk melihat semua variabel yang belum terselesaikan dalam kode yang dikonversi.
 - c. Pilih Gunakan fungsi konversi string dengan lebar tetap dari paket ekstensi untuk menggunakan fungsi paket ekstensi dalam kode SQL yang dikonversi. AWS SCT termasuk file paket ekstensi dalam proyek aplikasi Anda.
 - d. Pilih Ubah blok PL/SQL anonim menjadi panggilan SQL mandiri atau fungsi tersimpan untuk membuat prosedur tersimpan dalam database target Anda untuk semua blok PL/SQL anonim. AWS SCT kemudian termasuk menjalankan prosedur yang disimpan ini dalam kode aplikasi yang dikonversi.
 - e. Pilih Gunakan alur kursor khusus untuk meningkatkan konversi kursor database Oracle.
3. Di panel kiri, pilih menu konteks (klik kanan).
 4. Pilih Mengkonversi. AWS SCT menganalisis file kode sumber Anda, menentukan logika aplikasi, dan memuat metadata kode ke dalam proyek. Metadata kode ini mencakup kelas Pro*C, objek, metode, variabel global, antarmuka, dan sebagainya.

Di panel basis data yang akan AWS SCT membuat struktur folder yang mirip dengan proyek aplikasi sumber Anda. Di sini Anda dapat meninjau kode aplikasi yang dikonversi.



The screenshot displays two panels of SQL code. The top panel, titled 'Source Oracle app function: main() int', shows the original Oracle PL/SQL code. The bottom panel, titled 'Target Amazon RDS for PostgreSQL app function: main() int', shows the converted PostgreSQL code. The conversion process replaces Oracle-specific constructs like 'EXEC SQL' and 'return' with standard SQL statements.

```
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;
```


5. Simpan kode aplikasi yang telah dikonversi. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan kode aplikasi Anda yang telah dikonversi](#).

Mengedit dan menyimpan kode aplikasi Anda yang dikonversi dengan AWS SCT

Anda dapat mengedit pernyataan SQL yang dikonversi dan menggunakan AWS SCT untuk menyematkan kode yang diedit ini ke dalam kode aplikasi Pro* C yang dikonversi. Gunakan prosedur berikut untuk mengedit kode SQL yang telah dikonversi.

Untuk mengedit kode SQL Anda yang telah dikonversi

1. Perluas Pro* C simpul di bawah Aplikasi di panel kiri.
2. Pilih aplikasi untuk mengonversi, dan pilih menu konteks (klik kanan), dan pilih Mengkonversi.
3. Pada Lihat menu, pilih Tampilan laporan penilaian.
4. Pilih Simpan pernyataan ke CSV untuk menyimpan kode SQL yang diekstrak dari aplikasi Pro* C Anda sebagai file CSV.
5. Masukkan nama file CSV untuk menyimpan kode SQL yang diekstraksi, dan pilih Simpan.
6. Edit kode SQL yang diekstraksi.
7. Pada Lihat menu, pilih Tampilan utama.
8. Perluas Pro* C simpul di bawah Aplikasi di panel database yang akan dibangun.
9. Pilih aplikasi Anda yang dikonversi, dan pilih menu konteks (klik kanan), dan pilih Impor pernyataan dari CSV.
10. Pilih ya, lalu pilih file dengan kode SQL yang diedit, dan pilih Buka.

AWS SCT memecah pernyataan SQL yang dikonversi menjadi beberapa bagian dan menempatkannya ke objek yang sesuai dari kode aplikasi sumber Anda. Gunakan prosedur berikut untuk menyimpan kode aplikasi Anda yang telah dikonversi.

Untuk menyimpan kode aplikasi yang telah dikonversi

1. Perluas Pro* C simpul di bawah Aplikasi di panel database yang akan dibangun.
2. Pilih aplikasi Anda yang dikonversi, dan pilih Simpan.
3. Masukkan path ke folder untuk menyimpan kode aplikasi yang dikonversi, dan pilih Pilih folder.

Mengelola proyek konversi aplikasi Pro* C diAWS SCT

Anda dapat menambahkan beberapa proyek konversi aplikasi Pro* C, memperbarui kode aplikasi diAWS SCTproyek, atau hapus proyek konversi Pro* C dari proyek AndaAWS SCTproyek

Untuk menambahkan proyek konversi aplikasi Pro* C tambahan

1. PerluasAplikasisimpul di panel kiri.
2. PilihPro* Csimpul, dan buka menu konteks.
3. PilihAplikasi baru.
4. Masukkan informasi yang diperlukan untuk membuat proyek konversi aplikasi Pro*C baru. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi Pro* C](#).

Setelah Anda membuat perubahan dalam kode aplikasi sumber Anda, unggah keAWS SCTproyek

Untuk mengunggah kode aplikasi yang akan diperbarui

1. PerluasPro* Csimpul di bawahAplikasidi panel kiri.
2. Pilih aplikasi untuk memperbarui, dan buka menu konteks (klik kanan).
3. PilihSegarkandan kemudian memilihya.

AWS SCTmengunggah kode aplikasi Anda dari file sumber dan menghapus hasil konversi. Untuk menyimpan perubahan kode yang Anda buatAWS SCTdan hasil konversi, buat proyek konversi Pro* C baru.

Untuk menghapus proyek konversi aplikasi Pro* C

1. PerluasPro* Csimpul di bawahAplikasidi panel kiri.
2. Pilih aplikasi untuk menghapus, dan buka menu konteks (klik kanan).
3. PilihHapusdan kemudian memilihOKE.

Membuat laporan penilaian konversi aplikasi Pro*C diAWS SCT

TheLaporan penilaian konversi aplikasi Pro* Cmemberikan informasi tentang mengonversi kode SQL yang disematkan dalam aplikasi Pro* C Anda ke format yang kompatibel dengan basis data target Anda. Laporan penilaian memberikan rincian konversi untuk semua titik eksekusi SQL dan semua file

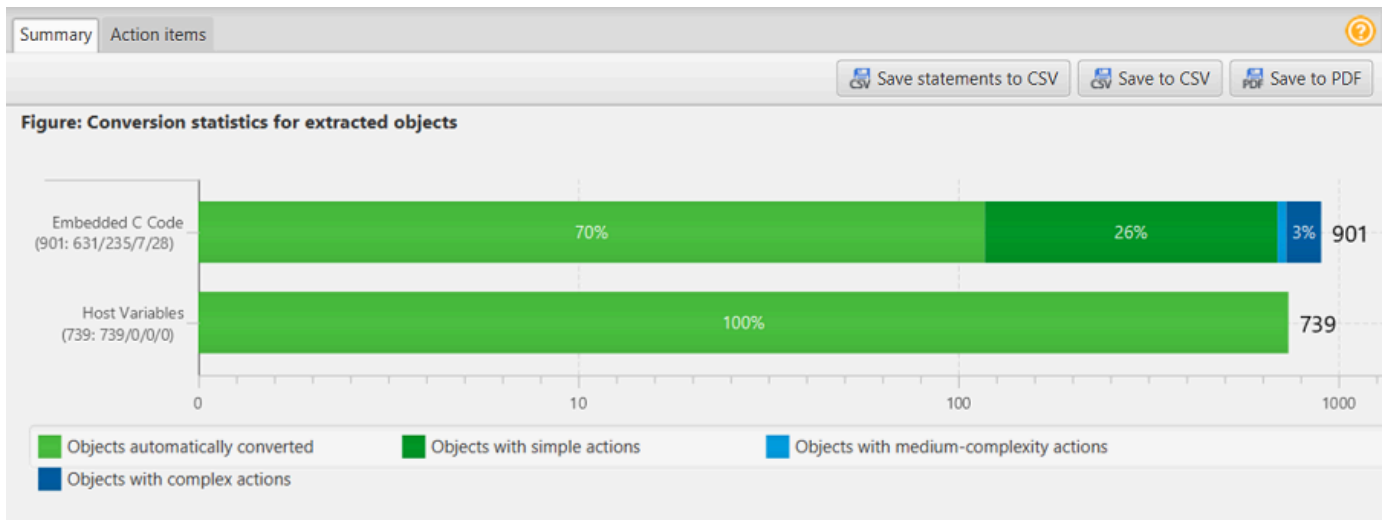
kode sumber. Laporan penilaian juga mencakup item tindakan untuk kode SQL yang AWS SCT tidak bisa mengkonversi.

Gunakan prosedur berikut untuk membuat laporan penilaian konversi aplikasi Pro*C.

Untuk membuat laporan penilaian konversi aplikasi Pro*C

1. Perluas Pro*C simpul di bawah Aplikasi di panel kiri.
2. Pilih aplikasi untuk mengonversi, dan buka menu konteks (klik kanan).
3. Pilih Mengkonversi.
4. Pada Lihat menu, pilih Tampilan laporan penilaian.
5. Tinjau Ringkasan Tinjau

The Ringkasan tab, ditampilkan berikut, menampilkan informasi ringkasan eksekutif dari laporan penilaian aplikasi Pro*C. Ini menunjukkan hasil konversi untuk semua titik eksekusi SQL dan semua file kode sumber.



6. Pilih Simpan pernyataan ke CSV untuk menyimpan kode SQL yang diekstrak dari aplikasi Pro*C Anda sebagai file nilai yang dipisahkan koma (CSV).
7. (Opsional) Simpan salinan lokal sebagai file PDF atau file nilai yang dipisahkan koma (CSV):
 - Pilih Simpan ke PDF di kanan atas untuk menyimpan laporan.

File PDF berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi aplikasi.
 - Pilih Simpan ke CSV di kanan atas untuk menyimpan laporan.

File CSV berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi kode SQL.

Menggunakan paket AWS SCT ekstensi

Paket AWS SCT ekstensi adalah modul add-on yang mengemulasi fungsi yang ada dalam database sumber yang diperlukan saat mengonversi objek ke database target. Sebelum Anda dapat menginstal paket AWS SCT ekstensi, Anda mengonversi skema database Anda.

Setiap paket AWS SCT ekstensi mencakup komponen-komponen berikut:

- Skema DB — Termasuk fungsi SQL, prosedur, dan tabel untuk meniru pemrosesan transaksi online tertentu (OLTP) dan objek database pemrosesan analitik online (OLAP) seperti urutan. Juga, mengemulasi tidak didukung built-in-functions dari database sumber. Nama skema ini memiliki format sebagai berikut: `aws_<database_engine_name>_ext`.
- AWS Lambda fungsi (untuk database OLTP tertentu) - Termasuk AWS Lambda fungsi yang meniru fungsionalitas database yang kompleks, seperti penjadwalan pekerjaan dan mengirim email.
- Pustaka khusus untuk database OLAP — Termasuk sekumpulan pustaka Java dan Python yang dapat Anda gunakan untuk memigrasi Microsoft SQL Server Integration Services (SSIS) mengekstrak, mengubah, dan memuat skrip (ETL) ke atau. AWS Glue AWS Glue Studio

Pustaka Java mencakup modul-modul berikut:

- `spark-excel_2.11-0.13.1.jar`— Untuk meniru fungsionalitas sumber Excel dan komponen target.
- `spark-xml_2.11-0.9.0.jar`, `poi-ooxml-schemas-4.1.2.jar`, dan `xmlbeans-3.1.0.jar` — Untuk meniru fungsionalitas komponen sumber XHTML.

Pustaka Python mencakup modul-modul berikut:

- `sct_utils.py`— Untuk meniru tipe data sumber dan menyiapkan parameter untuk kueri Spark SQL.
- `ssis_datetime.py`— Untuk meniru fungsi bawaan tanggal dan waktu.
- `ssis_null.py`— Untuk meniru ISNULL dan fungsi REPLACENULL bawaan.
- `ssis_string.py`- Untuk meniru fungsi bawaan string.

Untuk informasi lebih lanjut tentang perpustakaan ini, lihat [Menggunakan pustaka khusus untuk paket AWS SCT ekstensi](#).

Anda dapat menerapkan paket AWS SCT ekstensi dengan dua cara:

- AWS SCT dapat secara otomatis menerapkan paket ekstensi saat Anda menerapkan skrip basis data target dengan memilih Terapkan ke database dari menu konteks. AWS SCT menerapkan paket ekstensi sebelum menerapkan semua objek skema lainnya.
- Untuk menerapkan paket ekstensi secara manual, pilih database target dan kemudian pilih Terapkan paket ekstensi untuk dari menu konteks (klik kanan). Dalam kebanyakan situasi, aplikasi otomatis sudah cukup. Namun, Anda mungkin ingin menerapkan paket secara manual jika tidak sengaja dihapus.

Setiap kali Anda menerapkan paket AWS SCT ekstensi ke penyimpanan data target, komponen akan ditimpa, dan AWS SCT menampilkan pemberitahuan tentang hal ini. Untuk mematikan notifikasi ini, pilih Pengaturan, Pengaturan global, Pemberitahuan, lalu pilih Sembunyikan peringatan penggantian paket ekstensi.

Untuk konversi dari Microsoft SQL Server ke PostgreSQL, Anda dapat menggunakan paket ekstensi SQL Server ke PostgreSQL. AWS SCT Paket ekstensi ini mengemulasi SQL Server Agent dan SQL Server Database Mail. Lihat informasi yang lebih lengkap di [Meniru SQL Server Agent di PostgreSQL dengan paket ekstensi](#) dan [Meniru SQL Server Database Mail di PostgreSQL dengan paket ekstensi](#).

Berikut ini, Anda dapat menemukan informasi lebih lanjut tentang bekerja dengan paket AWS SCT ekstensi.

Topik

- [Izin untuk menggunakan paket AWS SCT ekstensi](#)
- [Menggunakan skema paket ekstensi](#)
- [Menggunakan pustaka khusus untuk paket AWS SCT ekstensi](#)
- [Menggunakan AWS Lambda fungsi dari paket AWS SCT ekstensi](#)
- [Mengkonfigurasi fungsi untuk paket AWS SCT ekstensi](#)

Izin untuk menggunakan paket AWS SCT ekstensi

Paket AWS SCT ekstensi untuk Amazon Aurora mengemulasi pengiriman email, penjadwalan pekerjaan, antrian, dan operasi lainnya menggunakan fungsi. AWS Lambda Saat Anda menerapkan paket AWS SCT ekstensi ke basis data Aurora target Anda, AWS SCT buat peran baru AWS Identity and Access Management (IAM) dan kebijakan IAM sebaris. Selanjutnya, AWS SCT buat fungsi Lambda baru, dan konfigurasi cluster Aurora DB Anda untuk koneksi keluar. AWS Lambda Untuk

menjalankan operasi ini, pastikan Anda memberikan izin yang diperlukan berikut kepada pengguna IAM Anda:

- `iam:CreateRole`— untuk membuat peran IAM baru untuk AWS akun Anda.
- `iam:CreatePolicy`— untuk membuat kebijakan IAM baru untuk AWS akun Anda.
- `iam:AttachRolePolicy`— untuk melampirkan kebijakan yang ditentukan ke peran IAM Anda.
- `iam:PutRolePolicy`— untuk memperbarui dokumen kebijakan inline yang disematkan dalam peran IAM Anda.
- `iam:PassRole`— untuk meneruskan peran IAM yang ditentukan ke mesin aturan.
- `iam:TagRole`— untuk menambahkan tag ke peran IAM.
- `iam:TagPolicy`— untuk menambahkan tag ke kebijakan IAM.
- `lambda:ListFunctions`— untuk melihat daftar fungsi Lambda Anda.
- `lambda:ListTags`— untuk melihat daftar tag fungsi Lambda Anda.
- `lambda:CreateFunction`— untuk membuat fungsi Lambda baru.
- `rds:AddRoleToDBCluster`— untuk mengaitkan peran IAM dengan cluster Aurora DB Anda.

Paket AWS SCT ekstensi untuk Amazon Redshift mengemulasi fungsi basis gudang data sumber yang diperlukan saat menerapkan objek yang dikonversi ke Amazon Redshift. Sebelum Anda menerapkan kode yang dikonversi ke Amazon Redshift, Anda harus menerapkan paket ekstensi untuk Amazon Redshift. Untuk melakukannya, sertakan `iam:SimulatePrincipalPolicy` tindakan dalam kebijakan IAM Anda.

AWS SCT menggunakan IAM Policy Simulator untuk memeriksa izin yang diperlukan untuk menginstal paket ekstensi Amazon Redshift. Simulator Kebijakan IAM dapat menampilkan pesan kesalahan bahkan jika Anda telah mengkonfigurasi pengguna IAM Anda dengan benar. Ini adalah masalah yang diketahui dari IAM Policy Simulator. Selain itu, IAM Policy Simulator menampilkan pesan kesalahan ketika Anda tidak memiliki `iam:SimulatePrincipalPolicy` tindakan dalam kebijakan IAM Anda. Dalam kasus ini, Anda dapat mengabaikan pesan kesalahan dan menerapkan paket ekstensi menggunakan wizard paket ekstensi. Untuk informasi selengkapnya, lihat [Menerapkan paket ekstensi](#).

Menggunakan skema paket ekstensi

Saat Anda mengonversi database atau skema gudang data, AWS SCT tambahkan skema tambahan ke basis data target Anda. Skema ini mengimplementasikan fungsi sistem SQL dari database sumber

yang diperlukan saat menulis skema yang dikonversi ke database target Anda. Skema tambahan ini disebut skema paket ekstensi.

Skema paket ekstensi untuk database OLTP diberi nama sesuai dengan database sumber sebagai berikut:

- Server Microsoft SQL: `AWS_SQLSERVER_EXT`
- MySQL: `AWS_MYSQL_EXT`
- Oracle: `AWS_ORACLE_EXT`
- PostgreSQL: `AWS_POSTGRESQL_EXT`

Skema paket ekstensi untuk aplikasi gudang data OLAP diberi nama sesuai dengan penyimpanan data sumber sebagai berikut:

- Greenplum: `AWS_GREENPLUM_EXT`
- Server Microsoft SQL: `AWS_SQLSERVER_EXT`
- Netezza: `AWS_NETEZZA_EXT`
- Oracle: `AWS_ORACLE_EXT`
- Teradata: `AWS_TERADATA_EXT`
- Vertika: `AWS_VERTICA_EXT`

Menggunakan pustaka khusus untuk paket AWS SCT ekstensi

Dalam beberapa kasus, tidak AWS SCT dapat mengonversi fitur database sumber ke fitur yang setara dalam database target Anda. Paket AWS SCT ekstensi yang relevan berisi pustaka khusus yang meniru beberapa fungsionalitas basis data sumber di database target Anda.

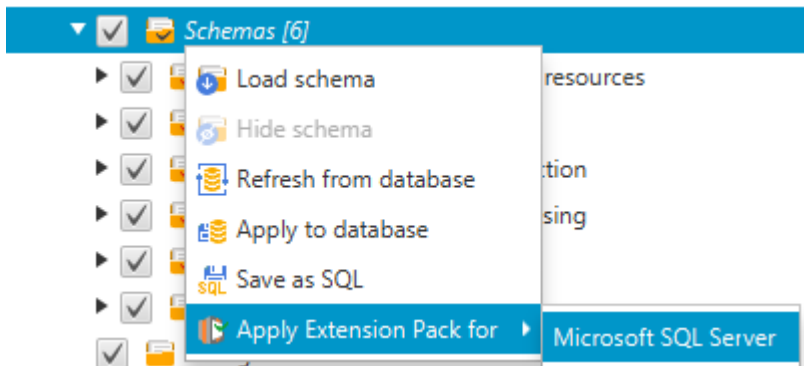
Jika Anda mengonversi database transaksional, lihat. [Menggunakan AWS Lambda fungsi dari paket AWS SCT ekstensi](#)

Menerapkan paket ekstensi

Anda dapat menerapkan paket AWS SCT ekstensi menggunakan wizard paket ekstensi atau ketika Anda menerapkan kode yang dikonversi ke database target Anda.

Untuk menerapkan paket ekstensi menggunakan wizard paket ekstensi

1. Di AWS Schema Conversion Tool, di pohon database target, buka menu konteks (klik kanan), pilih Terapkan paket ekstensi untuk, lalu pilih platform basis data sumber Anda.



Wizard paket ekstensi muncul.

2. Baca halaman Selamat Datang, lalu pilih Berikutnya.
3. Pada halaman pengaturan AWS profil, lakukan hal berikut:
 - Jika Anda menginstal ulang skema paket ekstensi saja, pilih Lewati langkah ini untuk saat ini, lalu pilih Berikutnya. Opsi Lewati langkah ini untuk saat ini hanya tersedia untuk database pemrosesan transaksi online (OLTP).
 - Jika Anda mengunggah perpustakaan baru, berikan kredensialnya untuk terhubung ke perpustakaan Anda. Akun AWS Gunakan langkah ini hanya ketika Anda mengonversi database OLAP atau skrip ETL. Anda dapat menggunakan kredensi AWS Command Line Interface (AWS CLI) Anda jika Anda telah menginstal. AWS CLI Anda juga dapat menggunakan kredensial yang sebelumnya Anda simpan di profil di pengaturan aplikasi global dan terkait dengan proyek. Jika perlu, pilih Navigasikan ke pengaturan global untuk mengonfigurasi kaitkan profil yang berbeda dengan AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Menyimpan profil AWS layanan di AWS SCT](#).
4. Jika Anda mengunggah perpustakaan baru, maka pilih Saya perlu mengunggah perpustakaan di halaman unggah Perpustakaan. Gunakan langkah ini hanya ketika Anda mengonversi database OLAP atau skrip ETL. Selanjutnya, berikan jalur Amazon S3, lalu pilih Upload library ke S3.

Jika Anda sudah mengunggah perpustakaan, lalu pilih Saya sudah mengunggah perpustakaan, gunakan bucket S3 saya yang ada di halaman unggah Perpustakaan. Selanjutnya, berikan jalur Amazon S3.

Setelah selesai, pilih Berikutnya.

5. Pada halaman Emulasi fungsi, pilih Buat paket ekstensi. Pesan muncul dengan status operasi paket ekstensi.

Setelah selesai, pilih Selesai.

Untuk menerapkan paket ekstensi saat menerapkan kode yang dikonversi

1. Tentukan bucket Amazon S3 di profil AWS layanan Anda. Gunakan langkah ini hanya ketika Anda mengonversi database OLAP atau skrip ETL. Untuk informasi selengkapnya, lihat [Menyimpan profil AWS layanan di AWS SCT](#).

Pastikan kebijakan bucket Amazon S3 Anda menyertakan izin berikut:

```
{
  "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"]
    }
  ]
}
```

Pada contoh sebelumnya, ganti *DataExtractionAgentName111122223333:user/* dengan nama pengguna IAM Anda.

2. Konversikan skema gudang data sumber Anda. Untuk informasi selengkapnya, lihat [Mengonversi skema gudang data ke Amazon Redshift](#).
3. Di panel kanan, pilih skema yang dikonversi.
4. Buka menu konteks (klik kanan) untuk elemen skema, lalu pilih Terapkan ke database.
5. AWS SCT menghasilkan paket ekstensi dengan komponen yang diperlukan dan menambahkan `aws_database_engine_name_ext` skema di pohon target. Selanjutnya, AWS SCT terapkan kode yang dikonversi dan skema paket ekstensi ke gudang data target Anda.

Saat Anda menggunakan kombinasi Amazon Redshift dan AWS Glue sebagai platform basis data target Anda, AWS SCT tambahkan skema tambahan dalam paket ekstensi.

Menggunakan AWS Lambda fungsi dari paket AWS SCT ekstensi

AWS SCT menyediakan paket ekstensi yang berisi fungsi Lambda untuk email, penjadwalan pekerjaan, dan fitur lain untuk database yang dihosting di Amazon EC2.

Menggunakan AWS Lambda fungsi untuk meniru fungsionalitas database

Dalam beberapa kasus, fitur database tidak dapat dikonversi ke fitur Amazon RDS yang setara. Misalnya, Oracle mengirim panggilan email yang digunakan `UTL_SMTP`, dan Microsoft SQL Server dapat menggunakan penjadwal pekerjaan. Jika Anda meng-host dan mengelola sendiri database di Amazon EC2, Anda dapat meniru fitur-fitur ini dengan AWS mengganti layanan untuk mereka.

Wisaya paket AWS SCT ekstensi membantu Anda menginstal, membuat, dan mengonfigurasi fungsi Lambda untuk meniru email, penjadwalan pekerjaan, dan fitur lainnya.

Menerapkan paket ekstensi untuk mendukung fungsi Lambda

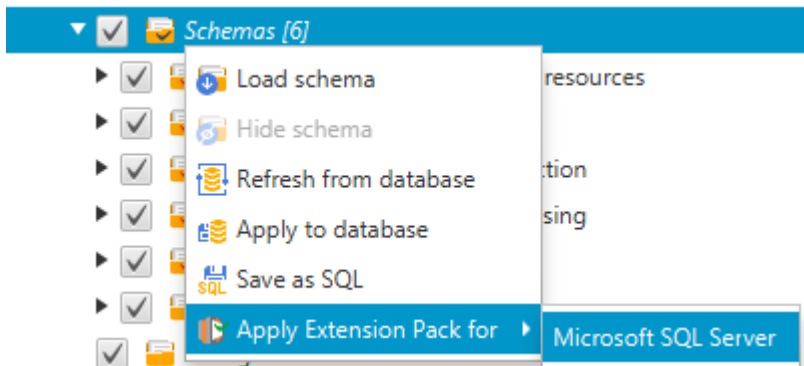
Anda dapat menerapkan paket ekstensi untuk mendukung fungsi Lambda menggunakan wizard paket ekstensi atau saat Anda menerapkan kode yang dikonversi ke basis data target Anda.

Important

Fitur emulasi AWS layanan hanya didukung untuk database yang diinstal dan dikelola sendiri di Amazon EC2. Jangan menginstal fitur emulasi layanan jika database target Anda ada di instans Amazon RDS DB.

Untuk menerapkan paket ekstensi menggunakan wizard paket ekstensi

1. Di AWS Schema Conversion Tool, di pohon database target, buka menu konteks (klik kanan), pilih Terapkan paket ekstensi untuk, lalu pilih platform basis data sumber Anda.



Wizard paket ekstensi muncul.

2. Baca halaman Selamat Datang, lalu pilih Berikutnya.
3. Pada halaman pengaturan AWS profil, lakukan hal berikut:
 - Jika Anda menginstal ulang skema paket ekstensi saja, pilih Lewati langkah ini untuk saat ini, lalu pilih Berikutnya.
 - Jika Anda menginstal AWS layanan, berikan kredensial untuk terhubung ke layanan Anda. Akun AWS Anda dapat menggunakan AWS CLI kredensial Anda jika Anda telah menginstal. AWS CLI Anda juga dapat menggunakan kredensial yang sebelumnya Anda simpan di profil di pengaturan aplikasi global dan terkait dengan proyek. Jika perlu, pilih Navigasikan ke Pengaturan Proyek untuk mengaitkan profil yang berbeda dengan proyek. Jika perlu, pilih Pengaturan Global untuk membuat profil baru. Untuk informasi selengkapnya, lihat [Menyimpan profil AWS layanan di AWS SCT](#).
4. Pada halaman Layanan Pengiriman Email, lakukan hal berikut:
 - Jika Anda menginstal ulang skema paket ekstensi saja, pilih Lewati langkah ini untuk saat ini, lalu pilih Berikutnya.
 - Jika Anda menginstal AWS layanan dan Anda memiliki fungsi Lambda yang ada, Anda dapat menyediakannya. Jika tidak, wizard membuatnya untuk Anda. Setelah selesai, pilih Berikutnya.
5. Pada halaman Job Emulation Service, lakukan hal berikut:
 - Jika Anda menginstal ulang skema paket ekstensi saja, pilih Lewati langkah ini untuk saat ini, lalu pilih Berikutnya.

- Jika Anda menginstal AWS layanan dan Anda memiliki fungsi Lambda yang ada, Anda dapat menyediakannya. Jika tidak, wizard membuatnya untuk Anda. Setelah selesai, pilih Berikutnya.
6. Pada halaman Emulasi fungsi, pilih Buat paket ekstensi. Pesan muncul dengan status operasi paket ekstensi.

Setelah selesai, pilih Selesai.

Note

Untuk memperbarui paket ekstensi dan menimpa komponen paket ekstensi lama, pastikan Anda menggunakan versi terbaru. AWS SCT Untuk informasi selengkapnya, lihat [Menginstal, memverifikasi, dan memperbarui AWS SCT](#).

Mengkonfigurasi fungsi untuk paket AWS SCT ekstensi

Paket ekstensi berisi fungsi yang harus Anda konfigurasi sebelum digunakan. Konstanta `CONVERSION_LANG` mendefinisikan bahasa yang digunakan paket layanan. Fungsinya tersedia untuk bahasa Inggris dan Jerman.

Untuk mengatur bahasa ke bahasa Inggris atau Jerman, buat perubahan berikut dalam kode fungsi. Temukan deklarasi konstan berikut:

```
CONVERSION_LANG CONSTANT VARCHAR := '';
```

Untuk mengatur `CONVERSION_LANG` ke bahasa Inggris, ubah baris menjadi berikut:

```
CONVERSION_LANG CONSTANT VARCHAR := 'English';
```

Untuk mengatur `CONVERSION_LANG` ke bahasa Jerman, ubah baris menjadi berikut:

```
CONVERSION_LANG CONSTANT VARCHAR := 'Deutsch';
```

Tetapkan pengaturan ini untuk fungsi-fungsi berikut:

- `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`

Praktik terbaik AWS SCT

Temukan informasi tentang praktik terbaik dan opsi untuk menggunakan AWS Schema Conversion Tool (AWS SCT).

Mengkonfigurasi memori tambahan

Untuk mengkonversi skema database yang besar, seperti database dengan 3.500 prosedur yang disimpan, Anda dapat mengkonfigurasi jumlah memori yang tersedia untuk AWS Schema Conversion Tool.

Untuk memodifikasi jumlah memori yang AWS SCT konsumsi

1. Pada menu Pengaturan, pilih Pengaturan global, lalu pilih opsi JVM.
2. Pilih Edit file konfigurasi dan pilih editor teks untuk membuka file konfigurasi.
3. Edit `JavaOptions` bagian untuk mengatur memori minimum dan maksimum yang tersedia. Contoh berikut menetapkan minimum untuk empat GB dan maksimum untuk 40 GB.

```
[JavaOptions]
-Xmx40960M
-Xms4096M
```

Kami menganjurkan Anda untuk mengatur memori minimum yang tersedia untuk setidaknya empat GB.

4. Simpan file konfigurasi, pilih OK, dan mulai ulang AWS SCT untuk menerapkan perubahan.

Mengkonfigurasi folder proyek default

AWS SCT menggunakan folder proyek untuk menyimpan file proyek, menyimpan laporan penilaian, dan menyimpan kode yang dikonversi. Secara default, AWS SCT menyimpan semua file di folder aplikasi. Anda dapat menentukan folder lain sebagai folder proyek default.

Mengubah folder proyek default

1. Pada menu Pengaturan, pilih Pengaturan global, lalu pilih Jalur file.
2. Untuk jalur file proyek Default, masukkan path ke folder proyek default.

3. Pilih Berlakukan, lalu pilih OKE.

Meningkatkan kecepatan migrasi data

Untuk memigrasi kumpulan data besar, seperti sekumpulan tabel dengan lebih dari 1 TB data, Anda mungkin ingin meningkatkan kecepatan migrasi. Saat Anda menggunakan agen ekstraksi data, kecepatan migrasi data bergantung pada berbagai faktor. Faktor-faktor ini mencakup jumlah irisan dalam kluster Amazon Redshift target Anda, ukuran file potongan dalam tugas migrasi Anda, RAM yang tersedia di PC tempat Anda menjalankan agen ekstraksi data, dan sebagainya.

Untuk meningkatkan kecepatan migrasi data, sebaiknya jalankan beberapa sesi migrasi pengujian dengan kumpulan data kecil dari data produksi Anda. Juga, kami menyarankan Anda menjalankan agen ekstraksi data Anda pada PC dengan SSD yang memiliki setidaknya 500 GB ukuran. Selama sesi pengujian ini, ubah parameter migrasi yang berbeda memantau pemanfaatan disk Anda untuk mengetahui konfigurasi yang memastikan kecepatan migrasi data maksimum. Kemudian, gunakan konfigurasi ini untuk memigrasi seluruh kumpulan data Anda.

Meningkatkan informasi logging

Anda dapat meningkatkan informasi logging yang dihasilkan oleh AWS SCT ketika mengkonversi database Anda, script, dan aplikasi SQL. Meskipun meningkatkan informasi pencatatan mungkin memperlambat konversi, perubahan tersebut dapat membantu Anda memberikan informasi yang kuat kepada AWS Support jika terjadi kesalahan.

AWS SCT menyimpan log dalam lingkungan lokal Anda. Anda dapat melihat file log ini dan membagikannya dengan AWS Support atau AWS SCT pengembang untuk pemecahan masalah.

Mengubah setelan log

1. Pada menu Pengaturan, pilih Pengaturan global, lalu pilih Logging.
2. Untuk jalur folder Log, masukkan folder untuk menyimpan log dari antarmuka pengguna.
3. Untuk jalur folder log konsol, masukkan folder untuk menyimpan log antarmuka baris perintah (CLI).
4. Untuk Ukuran file log maksimum (MB), masukkan ukuran, dalam MB, dari satu file log. Setelah file Anda mencapai batas ini, AWS SCT buat file log baru.
5. Untuk Jumlah maksimum file log, masukkan jumlah file log yang akan disimpan. Setelah jumlah file log dalam folder mencapai batas ini, AWS SCT hapus file log tertua.

6. Untuk jalur unduhan log Extractor, masukkan folder untuk menyimpan log agen ekstraksi data.
7. Untuk jalur log ekstraktor Cassandra, masukkan folder untuk menyimpan log agen ekstraksi data Apache Cassandra.
8. Pilih Minta jalur sebelum memuat untuk memastikan bahwa AWS SCT menanyakan tempat menyimpan log setiap kali Anda menggunakan agen ekstraksi data.
9. Untuk mode Debug, pilih Benar. Gunakan opsi ini untuk mencatat informasi tambahan saat AWS SCT log standar tidak menyertakan masalah apa pun.
10. Pilih modul aplikasi utama untuk meningkatkan informasi logging. Anda dapat meningkatkan informasi logging untuk modul aplikasi berikut:
 - Jenderal
 - Loader
 - Parser
 - Pencetak
 - Penyelesai
 - Telemetri
 - Konverter
 - Jenis pemetaan
 - Antarmuka pengguna
 - Controller
 - Bandingkan skema
 - Pusat data kloning
 - Penganalisis aplikasi

Untuk setiap modul aplikasi sebelumnya, pilih salah satu dari tingkat penebangan berikut:

- Jejak - Informasi paling rinci.
- Debug - Informasi rinci tentang aliran melalui sistem.
- Info - Peristiwa runtime, seperti startup atau shutdown.
- Peringatan - Penggunaan API yang tidak digunakan lagi, penggunaan API yang buruk, situasi waktu proses lainnya yang tidak diinginkan atau tidak terduga.
- Kesalahan - Kesalahan runtime atau kondisi yang tidak terduga.
- Kritis - Kesalahan yang menyebabkan aplikasi dimatikan.

- **Wajib** — Tingkat kesalahan tertinggi.

Secara default, setelah Anda mengaktifkan mode Debug, AWS SCT tetapkan level logging Info untuk semua modul aplikasi.

Misalnya, untuk membantu area masalah utama selama konversi, atur Parser, Type mapping, dan User interface ke Trace.

Jika informasi menjadi terlalu verbose untuk sistem file tempat log streaming, ubah ke lokasi dengan ruang yang cukup untuk menangkap log.

Untuk mengirimkan log ke AWS Support, buka direktori tempat log disimpan, dan kompres semua file ke dalam satu file.zip yang dapat dikelola. Kemudian unggah file.zip dengan kasus dukungan Anda. Ketika analisis awal selesai dan pengembangan yang sedang berlangsung dilanjutkan, kembalikan mode Debug ke false untuk menghilangkan penebangan verbose. Kemudian tingkatkan kecepatan konversi.

 **Tip**

Untuk mengelola ukuran log dan menyederhanakan masalah pelaporan, hapus log atau pindahkan ke lokasi lain setelah konversi berhasil. Melakukan tugas ini memastikan bahwa hanya kesalahan dan informasi yang relevan yang ditransmisikan ke AWS Support dan menjaga sistem file log agar tidak terisi.

Pemecahan masalah denganAWS SCT

Berikut ini, Anda dapat menemukan informasi tentang pemecahan masalah denganAWS Schema Conversion Tool (AWS SCT).

Tidak dapat memuat objek dari sumber basis data Oracle

Ketika Anda mencoba untuk memuat skema dari basis data Oracle, Anda mungkin mengalami salah satu dari kesalahan berikut.

```
Cannot load objects tree.
```

```
ORA-00942: table or view does not exist
```

Kesalahan ini terjadi karena pengguna yang ID-nya Anda gunakan untuk terhubung ke database Oracle tidak memiliki izin yang cukup untuk membaca skema, seperti yang dipersyaratkan olehAWS SCT.

Anda dapat menyelesaikan masalah dengan memberikan izin pengguna dan jugaselect_catalog_role izin untuk kamus apa pun dalam database. Izin ini menyediakan akses hanya-baca ke tampilan dan tabel sistem yang diperlukan olehAWS SCT. Contoh berikut membuat ID pengguna bernama min_privs dan memberikan pengguna dengan ID ini izin minimum yang diperlukan untuk mengkonversi skema dari database sumber Oracle.

```
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;
```

Pesan peringatan laporan penilaian

Untuk menilai kompleksitas konversi ke mesin database lain,AWS SCT memerlukan akses ke objek dalam database sumber Anda. KetikaAWS SCT mengalami masalah selama pemindaian dan tidak dapat melakukan penilaian, pesan peringatan dikeluarkan. Pesan ini menunjukkan bahwa persentase konversi keseluruhan berkurang. Berikut ini adalah alasan mengapaAWS SCT mungkin mengalami masalah selama pemindaian:

- Pengguna basis data Anda tidak memiliki akses ke semua objek yang dibutuhkan. Untuk informasi selengkapnya tentang izin keamanan dan hak istimewa yang AWS SCT diperlukan untuk database Anda, lihat [Sumber untuk AWS SCT](#) bagian database sumber yang sesuai dalam panduan ini.
- Objek yang dikutip dalam skema tidak ada lagi dalam database. Untuk membantu mengatasi masalah ini, Anda dapat terhubung dengan izin SYSDBA dan memeriksa apakah objek hadir dalam database.
- SCT sedang mencoba menilai objek yang dienkrpsi.

AWS SCT Referensi CLI

Bagian ini menjelaskan cara memulai dengan antarmuka baris AWS SCT perintah (CLI). Juga, bagian ini memberikan informasi tentang perintah utama dan mode penggunaan. Untuk referensi lengkap perintah AWS SCT CLI, lihat. [Bahan referensi](#)

Topik

- [Prasyarat untuk menggunakan antarmuka baris perintah AWS SCT](#)
- [AWS SCT Modus interaktif CLI](#)
- [Mendapatkan AWS SCT skenario CLI](#)
- [Mengedit AWS SCT skenario CLI](#)
- [AWS SCT Modus skrip CLI](#)
- [AWS SCT Bahan referensi CLI](#)

Prasyarat untuk menggunakan antarmuka baris perintah AWS SCT

Unduh dan instal versi terbaru Amazon Corretto 11. Untuk informasi selengkapnya, lihat [Unduhan untuk Amazon Corretto 11 di Panduan Pengguna](#) Amazon Corretto 11.

Unduh dan instal versi terbaru AWS SCT. Untuk informasi selengkapnya, lihat [Instalasi AWS SCT](#).

AWS SCT Modus interaktif CLI

Anda dapat menggunakan antarmuka AWS SCT baris perintah dalam mode interaktif. Dalam mode ini, Anda memasukkan perintah ke konsol satu per satu. Anda dapat menggunakan mode interaktif ini untuk mempelajari lebih lanjut tentang perintah CLI atau mengunduh skenario CLI yang paling umum digunakan.

Untuk mengonversi skema basis data sumber Anda AWS SCT, jalankan operasi urutan: buat proyek baru, sambungkan ke basis data sumber dan target, buat aturan pemetaan, dan konversi objek database. Karena alur kerja ini bisa rumit, sebaiknya gunakan skrip dalam mode CLI AWS SCT . Untuk informasi selengkapnya, lihat [Mode skrip](#).

Anda dapat menjalankan perintah AWS SCT CLI dari app folder jalur AWS SCT instalasi Anda. Di Windows, jalur instalasi default adalah `C:\Program Files\AWS Schema Conversion Tool\`. Pastikan folder ini menyertakan `AWSSchemaConversionToolBatch.jar` file.

Untuk masuk ke mode interaktif AWS SCT CLI, gunakan perintah berikut setelah Anda menyelesaikan prasyarat.

```
java -jar AWSSchemaConversionToolBatch.jar -type interactive
```

Sekarang Anda dapat menjalankan perintah AWS SCT CLI. Pastikan Anda mengakhiri perintah Anda / dengan baris baru. Juga, pastikan bahwa Anda menggunakan tanda kutip tunggal lurus (') sebelum dan sesudah nilai parameter perintah.

Note

Jika perintah sebelumnya kembaliUnexpected error, coba yang berikut ini:

```
java -Djdk.jar.maxSignatureFileSize=200000000 -jar  
AWSSchemaConversionToolBatch.jar
```

Untuk melihat daftar perintah yang tersedia dalam mode interaktif AWS SCT CLI, jalankan perintah berikut.

```
help  
/
```

Untuk melihat informasi tentang perintah AWS SCT CLI, gunakan perintah berikut.

```
help -command: 'command_name'  
/
```

Pada contoh sebelumnya, ganti *command_name* dengan *nama* perintah.

Untuk melihat informasi tentang parameter perintah AWS SCT CLI, gunakan perintah berikut.

```
help -command: 'command_name' -parameters: 'parameters_list'  
/
```

Pada contoh sebelumnya, ganti *command_name* dengan *nama* perintah. Kemudian, ganti *parameters_list* dengan *daftar* nama parameter yang dipisahkan dengan koma.

Untuk menjalankan skrip dari file dalam mode interaktif AWS SCT CLI, gunakan perintah berikut.

```
ExecuteFile -file: 'file_path'  
/
```

Pada contoh sebelumnya, ganti *file_path dengan path* ke file Anda dengan skrip. Pastikan file Anda memiliki `.scts` ekstensi.

Untuk keluar dari mode interaktif AWS SCT CLI, jalankan perintah. `quit`

Contoh-contoh

Contoh berikut menampilkan informasi tentang `Convert` perintah.

```
help -command: 'Convert'  
/
```

Contoh berikut menampilkan informasi tentang dua parameter `Convert` perintah.

```
help -command: 'Convert' -parameters: 'filter, treePath'  
/
```

Mendapatkan AWS SCT skenario CLI

Untuk mendapatkan AWS SCT skenario yang paling umum digunakan, Anda dapat menggunakan `GetCliScenario` perintah. Anda dapat menjalankan perintah ini dalam mode interaktif, lalu mengedit templat yang diunduh. Gunakan file yang diedit dalam mode skrip.

`GetCliScenarioPerintah` menyimpan template yang dipilih atau semua templat yang tersedia ke direktori yang ditentukan. Template berisi set lengkap perintah untuk menjalankan skrip. Pastikan Anda mengedit jalur file, kredensi database, nama objek, dan data lain dalam templat ini. Selain itu, pastikan Anda menghapus perintah yang tidak Anda gunakan dan tambahkan perintah baru ke skrip jika diperlukan.

Untuk menjalankan `GetCliScenario` perintah, selesaikan prasyarat dan masukkan mode interaktif CLI AWS SCT . Untuk informasi selengkapnya, lihat [Mode interaktif](#).

Selanjutnya, gunakan sintaks berikut untuk menjalankan `GetCliScenario` perintah dan mendapatkan AWS SCT skenario.

```
GetCliScenario -type: 'template_type' -directory: 'file_path'
```

/

Pada contoh sebelumnya, ganti *template_type* dengan *salah satu jenis* template dari tabel berikut. Selanjutnya, ganti *file_path* dengan path folder tempat Anda ingin mengunduh skrip. Pastikan bahwa AWS SCT dapat mengakses folder ini tanpa meminta hak admin. Juga, pastikan bahwa Anda menggunakan tanda kutip tunggal (') sebelum dan sesudah nilai parameter perintah.

Untuk mengunduh semua template AWS SCT CLI, jalankan perintah sebelumnya tanpa opsi. -type

Tabel berikut mencakup jenis template AWS SCT CLI yang dapat Anda unduh. Untuk setiap template, tabel menyertakan nama file dan deskripsi operasi yang dapat Anda jalankan menggunakan skrip.

Jenis template	Nama file	Deskripsi
BTEQ ScriptConversion	BTEQScriptConversionTemplate.scts	Mengonversi Teradata Basic Teradata Query (BTEQ),, FastExport dan FastLoad skrip ke Amazon Redshift RSQL. MultiLoad Untuk informasi selengkapnya, lihat Mengonversi proses ETL.
ConversionApply	ConversionTemplate.scts	Mengkonversi skema database sumber dan menerapkan kode dikonversi ke database target. Secara opsional, menyimpan kode yang dikonversi sebagai skrip SQL, dan menyimpan laporan penilaian. Untuk informasi selengkapnya, lihat Mengonversi skema database.
GenericAppConversion	GenericApplicationConversionTemplate.scts	Mengonversi kode SQL yang disematkan ke dalam aplikasi Anda dengan konverter

Jenis template	Nama file	Deskripsi
		aplikasi generik AWS SCT . Untuk informasi selengkapnya, lihat Mengonversi kode SQL dalam aplikasi Anda .
HadoopMigration	HadoopMigrationTemplate.scts	Memigrasi klaster Hadoop lokal Anda ke Amazon EMR. Untuk informasi selengkapnya, lihat Menggunakan Apache Hadoop sebagai sumber untuk AWS SCT .
HadoopResumeMigration	HadoopResumeMigrationTemplate.scts	Melanjutkan migrasi terputus dari klaster Hadoop lokal Anda ke Amazon EMR. Untuk informasi selengkapnya, lihat Menggunakan Apache Hadoop sebagai sumber untuk AWS SCT .
Informatika	InformaticaConversionTemplate.scts	Mengonversi kode SQL yang disematkan ke dalam skrip ekstrak, transformasi, dan muat (ETL) Informatica Anda. Mengonfigurasi koneksi ke basis data sumber dan target Anda dalam skrip ETL Anda, dan menyimpan skrip yang dikonversi setelah konversi. Untuk informasi selengkapnya, lihat Mengkonversi skrip ETL Informatica .

Jenis template	Nama file	Deskripsi
LanguageSpecificAppConversion	LanguageSpecificAppConversionTemplate.scts	Mengonversi kode SQL yang disematkan ke dalam aplikasi C #, C ++, Java, dan Pro*C Anda dengan konverter aplikasi. AWS SCT Untuk informasi selengkapnya, lihat Mengonversi aplikasi SQL .
OozieConversion	OozieConversionTemplate.scts	Mengonversi alur kerja Apache Oozie Anda menjadi AWS Step Functions Untuk informasi selengkapnya, lihat Menggunakan Apache Oozie sebagai sumber untuk AWS SCT .
RedshiftAgent	DWHDataMigrationTemplate.scts	Mengonversi skema gudang data sumber dan menerapkan kode yang dikonversi ke database Amazon Redshift target. Kemudian mendaftarkan agen ekstraksi data, membuat dan memulai tugas migrasi data. Untuk informasi selengkapnya, lihat Migrasi dari gudang data ke Amazon Redshift .
ReportCreation	ReportCreationTemplate.scts	Membuat laporan migrasi database untuk beberapa skema database sumber. Kemudian simpan laporan ini sebagai CSV file PDF. Untuk informasi selengkapnya, lihat Laporan penilaian migrasi .

Jenis template	Nama file	Deskripsi
SQL ScriptConversion	SQLScriptConversionTemplate.scts	Mengonversi skrip SQL* Plus atau TSQL ke PL/SQL dan menyimpan skrip yang dikonversi. Juga, menyimpan laporan penilaian.

Setelah Anda mengunduh template AWS SCT CLI, gunakan editor teks untuk mengonfigurasi skrip agar berjalan di basis data sumber dan target Anda. Selanjutnya, gunakan mode skrip AWS SCT CLI untuk menjalankan skrip Anda. Untuk informasi selengkapnya, lihat [AWS SCT Modus skrip CLI](#).

Contoh-contoh

Contoh berikut mengunduh semua template ke dalam C:\SCT\Templates folder.

```
GetCliScenario -directory: 'C:\SCT\Templates'  
/
```

Contoh berikut mengunduh template untuk ConversionApply operasi ke dalam C:\SCT\Templates folder.

```
GetCliScenario -type: 'ConversionApply' -directory: 'C:\SCT\Templates'  
/
```

Mengedit AWS SCT skenario CLI

Setelah Anda mengunduh template skenario, konfigurasi mereka untuk mendapatkan skrip yang berfungsi yang dapat berjalan di database Anda.

Untuk semua templat, pastikan Anda memberikan jalur ke driver untuk basis data sumber dan target Anda. Untuk informasi selengkapnya, lihat [Mengunduh driver database yang diperlukan](#).

Pastikan Anda menyertakan kredensi database untuk basis data sumber dan target. Selain itu, pastikan Anda menyiapkan aturan pemetaan untuk menjelaskan pasangan sumber-target untuk proyek konversi Anda. Untuk informasi selengkapnya, lihat [Membuat aturan pemetaan](#).

Selanjutnya, konfigurasi ruang lingkup operasi yang akan dijalankan. Anda dapat menghapus perintah yang tidak Anda gunakan atau menambahkan perintah baru ke skrip.

Misalnya, Anda berencana untuk mengonversi semua skema di database Oracle sumber Anda ke PostgreSQL. Kemudian Anda berencana untuk menyimpan laporan penilaian migrasi database Anda sebagai PDF dan menerapkan kode yang dikonversi ke database target. Dalam hal ini, Anda dapat menggunakan template untuk `ConversionApply` operasi. Gunakan prosedur berikut untuk mengedit template AWS SCT CLI Anda.

Untuk mengedit template AWS SCT CLI untuk operasi **ConversionApply**

1. Buka `ConversionTemplate.scts` yang Anda unduh. Untuk informasi selengkapnya, lihat [Contoh-contoh](#).
2. Hapus `CreateFilter`, `Konversi -filter`, `ApplyToTarget -filter`, `SaveTargetSQL`, `SaveTargetSQLbyStatement`, dan operasi `SaveReportCSV` dari skrip Anda.
3. Untuk `oracle_driver_file` dalam `SetGlobalSettingsoperasi`, masukkan path ke driver Oracle Anda. Kemudian, untuk `postgresql_driver_file`, masukkan path ke driver PostgreSQL Anda.

Jika Anda menggunakan mesin database lain, gunakan nama yang sesuai untuk pengaturan. Untuk daftar lengkap pengaturan global yang dapat Anda atur dalam `SetGlobalSettingsoperasi`, lihat Matriks pengaturan global di [Bahan referensi](#).

4. (Opsional) Untuk `CreateProject`, masukkan nama proyek Anda dan lokasi untuk file proyek lokal Anda. Jika Anda memilih untuk melanjutkan dengan nilai default, pastikan bahwa AWS SCT dapat membuat file di `C:\temp` folder tanpa meminta hak admin.
5. Untuk `AddSource`, masukkan alamat IP server basis data sumber Anda. Juga, masukkan nama pengguna, kata sandi, dan port untuk terhubung ke server basis data sumber Anda.
6. Untuk `AddTarget`, masukkan alamat IP server basis data target Anda. Juga, masukkan nama pengguna, kata sandi, dan port untuk terhubung ke server basis data target Anda.
7. (Opsional) Untuk `AddServerMapping`, masukkan objek database sumber dan target yang ingin Anda tambahkan ke aturan pemetaan. Anda dapat menggunakan `sourceTreePath` dan `targetTreePath` parameter untuk menentukan jalur ke objek database. Opsional, Anda dapat menggunakan `sourceNamePath` dan `targetNamePath` untuk menentukan nama-nama objek database. Untuk informasi selengkapnya, lihat Perintah pemetaan server di [Bahan referensi](#)

Nilai default `AddServerMappingoperasi` memetakan semua skema sumber dengan database target Anda.

8. Simpan file dan kemudian gunakan mode skrip untuk menjalankannya. Untuk informasi selengkapnya, lihat [Mode skrip](#).

AWS SCT Modus skrip CLI

Setelah Anda membuat skrip AWS SCT CLI atau mengedit template, Anda dapat menjalankannya dengan perintah. RunSCTBatch Pastikan Anda menyimpan file Anda dengan skrip CLI sebagai ekstensi `.scts`.

Anda dapat menjalankan skrip AWS SCT CLI dari app folder jalur instalasi Anda AWS SCT . Di Windows, jalur instalasi default adalah `C:\Program Files\AWS Schema Conversion Tool\`. Pastikan folder ini menyertakan `RunSCTBatch.sh` file `RunSCTBatch.cmd` atau. Juga, folder ini harus menyertakan `AWSSchemaConversionToolBatch.jar` file.

Atau, Anda dapat menambahkan path ke `RunSCTBatch` file dalam variabel `PATH` lingkungan pada sistem operasi Anda. Setelah Anda memperbarui variabel `PATH` lingkungan, Anda dapat menjalankan skrip AWS SCT CLI dari folder apa pun.

Untuk menjalankan skrip AWS SCT CLI, gunakan perintah berikut di Windows.

```
RunSCTBatch.cmd --pathtoscts "file_path"
```

Pada contoh sebelumnya, ganti *file_path dengan path* ke file Anda dengan skrip.

Untuk menjalankan skrip AWS SCT CLI, gunakan perintah berikut di Linux.

```
RunSCTBatch.sh --pathtoscts "file_path"
```

Pada contoh sebelumnya, ganti *file_path dengan path* ke file Anda dengan skrip.

Anda dapat memberikan parameter opsional dalam perintah ini, seperti kredensial database, tingkat detail dalam output konsol, dan lainnya. Untuk informasi lebih lanjut, unduh referensi antarmuka baris AWS SCT perintah di [Bahan referensi](#).

Contoh-contoh

Contoh berikut menjalankan `ConversionTemplate.scts` script dalam `C:\SCT\Templates` folder. Anda dapat menggunakan contoh ini di Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\SCT\Templates\ConversionTemplate.scts"
```

Contoh berikut menjalankan `ConversionTemplate.scts` script dalam `/home/user/SCT/Templates` direktori. Anda dapat menggunakan contoh ini di Linux.

```
RunSCTBatch.sh --pathtoscts "/home/user/SCT/Templates/ConversionTemplate.scts"
```

AWS SCT Bahan referensi CLI

[Anda dapat menemukan bahan referensi tentang antarmuka baris AWS Schema Conversion Tool perintah \(CLI\) dalam panduan berikut: Referensi CLI AWS Schema Conversion Tool .](#)

Catatan rilis untuk AWS SCT

Bagian ini berisi catatan rilis untuk AWS SCT, dimulai dengan versi 1.0.640.

Catatan rilis untuk AWS SCT Build 675

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Cassandra	DynamoD	Memperbaiki bug di mana instalasi Cassandra akan gagal pada Datacenter target.	Tidak
DB2 LUW	PostgreSQL	DYNAMIC SQL: PREPARE statement: Resolve dan Conversion tanpa Dynamic SQL.	Tidak
DB2 LUW	PostgreSQL	Menambahkan dukungan untuk REGISTER KHUSUS.	Tidak
DB2 LUW	PostgreSQL	Pembaruan paket ekstensi	Tidak
Hadoop	Amazon EMR	Menambahkan dukungan untuk menghubungkan ke cluster Hadoop melalui protokol rsa-sha2.	Tidak
Microsoft SQL Server	Amazon Redshift	Perbaiki untuk Driver JDBC yang memaksa TLS meskipun tidak dikonfigurasi.	Tidak
Netezza	Amazon Redshift	Menambahkan dukungan untuk konversi tampilan Terwujud.	Tidak
Oracle	Amazon Redshift	Menambahkan dukungan untuk kueri rekursif di Amazon Redshift.	Ya
Oracle	PostgreSQL, Aurora	Perbaiki untuk konversi tipe data NUMBER yang salah.	Ya

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
	PostgreSQL		
Oracle	Amazon Redshift	Migrasi data. Partisi otomatis Oracle. Ditambahkan waktu kedaluwarsa untuk nilai fragmen tabel. Waktu kedaluwarsa adalah 72 jam. Saat kedaluwarsa terjadi, fragmen data dibangun kembali saat tugas migrasi data dibuat.	Tidak
Oracle	Amazon Redshift	SCT Data Extractor: Mengubah pendekatan mengunggah data ke Amazon Redshift. Secara default, extractor tidak membuat tabel bertahap. Sebagai gantinya, setelah semua file data berada di bucket Amazon S3, ekstraktor menyalinnya ke tabel target menggunakan satu perintah COPY.	Tidak
Oracle	Amazon Redshift	Menambahkan migrasi tipe data RAW ke kolom VARBYTE.	Tidak
Oracle	PostgreSQL, Aurora PostgreSQL	Konversi multi-versi	Tidak
Oracle	PostgreSQL	Ditambahkan dukungan untuk Pernyataan MERGE di PostgreSQL 15.x.	Ya
Oracle	PostgreSQL	Ditambahkan dukungan untuk fungsi ekspresi reguler baru di PostgreSQL 15.x.	Ya

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Oracle	PostgreSQL, Aurora PostgreSQL	Pernyataan ON CONFLICT DO UPDATE dikonversi tanpa alias yang dikecualikan.	Ya
Teradata	Amazon Redshift	Menambahkan dukungan konversi untuk fungsi LEAD/LAG.	Tidak
Teradata	Amazon Redshift	Pengecoran tipe data yang disempurnakan dengan indikasi eksplisit format data.	Tidak
Teradata	Amazon Redshift	Peningkatan konversi klausa AT 'TIME ZONE' dalam ekspresi waktu/stempel waktu.	Tidak
Teradata	Amazon Redshift	AI-9996 selama prosedur konversi dengan pernyataan MERGE.	Tidak

Catatan rilis untuk AWS SCT Build 674

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Semua	Semua	Berbagai perbaikan bug dan peningkatan kinerja	Sebagian (hanya untuk pasangan sumber dan target yang didukung)
Azure SQL/ Microsoft	Amazon Redshift	Menghapus pesan "AI 18066: Tidak dapat mengonversi nama skema" yang menyesatkan pengguna selama penilaian/konversi skema	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
SQL Server			
Azure SQL/ Micro soft SQL Server	Amazon RDS untuk MySQL/ Amazon Aurora MySQL	Konversi prosedur yang salah tanpa menetapkan kode pengembalian	Sebagian (Konversi Skema saat ini tidak mendukung Azure SQL sebagai sumber)
Azure SQL/ Micro soft SQL Server	Amazon RDS untuk PostgreSQL/ Amazon Aurora PostgreSQL	Memperbaiki AI9997 untuk beberapa kasus konversi klausa FOR XMLPATH	Sebagian (Konversi Skema saat ini tidak mendukung Azure SQL sebagai sumber)
Azure SQL/ Micro soft SQL Server	Amazon RDS untuk PostgreSQL/ Amazon Aurora PostgreSQL	Nilai dibulatkan ke skala asli dalam prosedur/fungsi tubuh	Sebagian (Konversi Skema saat ini tidak mendukung Azure SQL sebagai sumber)

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Azure SQL/ Micro soft SQL Server	Amazon RDS untuk PostgreSQL/ Amazon Aurora PostgreSQL	Berbagai perbaikan konversi pernyataan EXECUTE	Sebagian (Konversi Skema saat ini tidak mendukung Azure SQL sebagai sumber)
Azure SQL/ Server Microsoft SQL/ Azure Synapse	Amazon Redshift	Peningkatan konversi dari pernyataan dan mode berikut: <ul style="list-style-type: none"> • BLOK PENGECEUALIAN • KOMIT OTOMATIS • NONATOMIK • SET PENGELOMPOKAN • KUBUS • MENGGULUNG 	Tidak
DB2 LUW	Amazon RDS untuk PostgreSQL/ Amazon Aurora PostgreSQL	Berbagai perbaikan dalam metadata memuat sql-query	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
DB2 LUW	Amazon RDS untuk PostgreSQL / Amazon Aurora PostgreSQL	AI 9996 tidak diharapkan pada pemicu	Tidak
DB2 z/OS	Amazon RDS untuk PostgreSQL / Amazon Aurora PostgreSQL	Fungsi analitik ROWNUMBER	Tidak
DB2 z/OS	Amazon RDS untuk PostgreSQL / Amazon Aurora PostgreSQL	Dukungan konstan string heksadesimal	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
DB2 z/OS	Amazon RDS untuk PostgreSQL/ Amazon Aurora PostgreSQL	Berbagai perbaikan dalam metadata memuat sql-query.	Tidak
DB2 z/OS	Amazon RDS untuk PostgreSQL/ Amazon Aurora PostgreSQL	NILAI BERIKUTNYA UNTUK dukungan referensi urutan	Tidak
DB2 z/OS	Amazon RDS untuk PostgreSQL/ Amazon Aurora PostgreSQL	GET DIAGNOSTICS pernyataan dukungan opsi DB2_NUMBER_ROWS	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
DB2 z/OS	Amazon RDS untuk PostgreSQL/ Amazon Aurora PostgreSQL	GET DIAGNOSTICS Beberapa pernyataan	Tidak
DB2 z/OS	Amazon RDS untuk PostgreSQL/ Amazon Aurora PostgreSQL	Memperbaiki bug dalam konversi pernyataan FOR.	Tidak
Oracle	Amazon RDS untuk MySQL/ Amazon Aurora MySQL	Memperbaiki bug saat node parameter fungsi paket tidak didefinisikan.	Ya

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Oracle	Amazon RDS untuk PostgreSQL / Amazon Aurora PostgreSQL	Memperbaiki bug dalam fungsi paket ekstensi AWS_ORACLE_EXT.NEXT_DAY	Ya
Oracle	Amazon RDS untuk PostgreSQL / Amazon Aurora PostgreSQL	Memperbaiki berbagai bug dengan konversi “(+)” di gabungan luar Oracle	Ya
Oracle		Support Otentikasi Kerberos	Tidak
SAP ASE	Amazon RDS untuk PostgreSQL / Amazon Aurora PostgreSQL	Memperbaiki bug saat mengonversi lebih dari satu pengenal dalam klausa FROM dalam pernyataan UPDATE	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
SAP ASE	Amazon RDS untuk PostgreSQL/ Amazon Aurora PostgreSQL	Memperbaiki bug dengan komentar multi-baris dan konversi pernyataan	Tidak
SAP ASE		Menambahkan dukungan untuk parameter ENCRYPT_PASSWORD saat menghubungkan	Tidak
Teradata	Amazon Redshift	Peningkatan konversi tabel VOLATILE dengan nama skema tertentu	Tidak
Teradata	Amazon Redshift	Konversi salah WHERE CLAUSE dalam CTE kompleks	Tidak
Teradata	Amazon Redshift	Menambahkan dukungan untuk tipe data INTERVAL saat memigrasikan data menggunakan Agen Ekstraksi Data SCT.	Tidak
Skrip Teradata BTEQ	Skrip RSQL Amazon Redshift	Parameter konversi keluar yang salah dalam prosedur yang dijalankan oleh BTEQ	Tidak

Catatan rilis untuk AWS SCT Build 673

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Perbaiki bug umum dan peningkatan kinerja

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Azure SQL/Micro soft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Memperbaiki konversi panggilan fungsi yang salah
Azure SQL/Micro soft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Konversi yang diimplementasikan dari FOR klausa XML
Azure SQL/Micro soft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Konversi klausa FOR XML dengan alias yang salah.
Azure SQL/Micro soft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Memperbaiki bug saat AWS SCT tidak mengonversi EXECUTE pernyataan yang menjalankan string karakter dengan parameter prosedur.
Azure SQL/Micro soft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Peningkatan konversi pernyataan UPDATE dengan gabungan batin.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Sinaps Azure	Amazon Redshift	Memperbaiki konversi fungsi OBJECT_ID bawaan yang salah.
IBM DB2 untuk z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	Menerapkan konversi pernyataan dan objek berikut: <ul style="list-style-type: none"> • DECLARE TEMPORARY TABLE statement • DROP TABLE statement • PK dan UNIQUE kendala pada tabel yang dipartisi • Fungsi TIMESTAMPDIFF • Fungsi TO_DATE • Fungsi EBCDIC_STR • Fungsi VARCHAR_FORMAT
IBM DB2 untuk z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	Memperbaiki bug saat indeks berbasis fungsi melewati fungsi setelah konversi.
IBM DB2 untuk z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	Memperbaiki bug di mana REPEAT pernyataan ditutup dengan AI 9996 setelah konversi
IBM DB2 untuk z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	Memperbaiki bug di mana FINAL TABLE klausa ditutup dengan 9996.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM DB2 untuk z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	LOADER Kunci partisi dalam batasan referensi. AWS SCT sekarang dapat mengonversi kunci primer dan kendala unik dalam tabel yang dipartisi sebagai indeks sekunder.
IBM DB2 untuk z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	Dukungan fungsi PostgreSQL.VARCHAR_Format
IBM DB2 untuk z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	Menerapkan perubahan pemeriksaan CreateTransformationRule dan perintah ModifyTransformationRule CLI SCT.
Greenplum	Amazon Redshift	Memperbaiki bug dengan panggilan prosedur tersimpan yang salah setelah konversi
Hadoop	Amazon EMR	Menambahkan dukungan untuk menghubungkan ke cluster Hadoop menggunakan protokol rsa-sha2.
Hadoop	Amazon EMR	Menambahkan dukungan untuk Amazon EMR dengan metastore Non-glue Hive,
Oracle	Amazon Redshift	Memperbaiki bug dengan konversi kueri rekursif yang salah di mana kolom PRIOR tidak ada dalam daftar SELECT.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Diimplementasikan mengembalikan elemen array asosiatif
Oracle	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Memperbaiki AI 9996 yang tidak terduga UNPIVOT dengan tanda kurung
Oracle	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Memperbaiki AI 9996 yang tidak terduga dengan UNPIVOT UNION ALL
Oracle	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	perbaikan untuk konversi tipe Number data
Oracle	Ekstraktor Data Pergeseran Merah Amazon	Support untuk partisi otomatis untuk tabel Oracle. Optimalisasi untuk membuat tugas migrasi.
Teradata	Amazon Redshift	Menerapkan konversi EXCEPTION BLOCK pernyataan

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Support untuk konversi ALL, ANY, dan SOME predikat ke Amazon Redshift.
Teradata	Amazon Redshift	Menambahkan dukungan asli untuk QUALIFY predikat.
Teradata	Amazon Redshift	Peningkatan konversi berikut ini: <ul style="list-style-type: none"> • Kueri rekursif • GROUPING SET • CUBE • ROLLUP • UPDATE pernyataan dengan gabungan implisit
Sumber OLAP	Ekstraktor Data Pergeseran Merah Amazon	Menerapkan perintah CLI untuk Stop/Resume untuk tugas Amazon Redshift Data Extractor.
Sumber OLAP	Ekstraktor Data Pergeseran Merah Amazon	Menambahkan kemampuan untuk memilih kolom tabel yang perlu dimigrasi selama konfigurasi tugas migrasi.

Catatan rilis untuk AWS SCT Build 672

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Amazon RDS for PostgreSQL	Menerapkan dukungan PostgreSQL mayor versi 15 sebagai target migrasi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Amazon Redshift	Menambahkan <code>PrintTaskStatus</code> perintah baru di antarmuka baris AWS SCT perintah (CLI) untuk menampilkan status tugas migrasi data.
Semua	Amazon Redshift	Meningkatkan aliran konfigurasi untuk agen ekstraksi data.
Semua	Amazon Redshift	Mengatasi kesalahan di mana agen ekstraksi data tidak menampilkan informasi tentang subtugas.
Apache Oozie	AWS Step Functions	Menambahkan opsi untuk menyimpan definisi mesin negara sebagai skrip dalam kode yang dikonversi.
Database Azure SQL Microsoft SQL Server	Aurora PostgreSQL	Menerapkan konversi <code>COALESCE</code> , <code>DATEADD</code> , <code>GETDATE</code> , dan <code>SUM</code> fungsi.
Database Azure SQL Microsoft SQL Server	Aurora PostgreSQL	Meningkatkan konversi <code>UPDATE</code> pernyataan dengan <code>JOIN</code> dan <code>OUTPUT</code> klausa.
Database Azure SQL Microsoft SQL Server	Aurora PostgreSQL	Menyelesaikan kesalahan yang terjadi selama konversi <code>SELECT TOP 1 WITH TIES</code> pernyataan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Database Azure SQL Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan beberapa masalah yang terjadi selama konversi FOR XML klausa dalam fungsi bawaan.
Greenplum	Amazon Redshift	Menerapkan konversi GET DIAGNOSTICS dan RAISE pernyataan dengan menggunakan blok Amazon Redshift EXCEPTION asli.
Greenplum	Amazon Redshift	Meningkatkan konversi prosedur tersimpan dengan menambahkan dukungan EXCEPTION blok dalam kode yang dikonversi.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Memperbaiki kesalahan di mana TO_CHAR fungsi dengan templat format waktu tidak dikonversi dengan benar.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menerapkan konversi ekspresi tabel bersarang.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menerapkan konversi GOTO, MERGE, REPEAT, dan SIGNAL pernyataan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menerapkan konversi FETCH pernyataan dengan BEFORE dan kata kunci AFTER orientasi.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menerapkan konversi referensi FINAL TABLE dan OLD TABLE tabel.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menerapkan konversi fungsi-fungsi berikut. <ul style="list-style-type: none"> • ADD_MONTHS • DAY dengan parameter tipe data karakter • 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 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi alias subquery dalam JOIN klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi COALESCE fungsi.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi EXPLICIT indeks.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi nama kolom dalam ekspresi majemuk untuk menyelesaikan masalah di mana item tindakan 9997 tiba-tiba muncul selama konversi.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi kunci primer dan kendala unik.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Memperbaiki konversi XMLTABLE pernyataan dalam INSERT pernyataan untuk menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi fungsi dengan argumen. SUBSTR
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi register khusus. CURRENT_TIMESTAMP
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah saat item tindakan 9996 muncul secara tak terduga selama konversi MERGE pernyataan, pernyataan yang tidak didukung, dan fungsi bawaan yang tidak didukung.
Microsoft SQL Server	Semua	Menambahkan dukungan Microsoft SQL Server versi 2022 sebagai sumber.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi SELECT pernyataan yang menggunakan operator penggabungan string. AWS SCT menggunakan STRING_AGG fungsi dalam kode yang dikonversi.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Menerapkan dukungan versi baru 3.1.0 dari file konfigurasi fitur Babelfish. File ini mendefinisikan fitur SQL yang didukung dan tidak didukung oleh versi Babelfish tertentu.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Netezza	Amazon Redshift	Menyelesaikan masalah di mana agen ekstraksi data tidak memulai migrasi data dari titik CDC yang ditentukan.
Oracle	Semua	Memperbarui laporan penilaian untuk database Oracle versi 19 sebagai sumber.
Oracle	Aurora PostgreSQL L PostgreSQL L	Menerapkan konversi DBMS_OUTPUT paket dengan menambahkan fungsi baru ke paket AWS SCT ekstensi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Menerapkan konversi fungsi dan prosedur yang menggunakan array asosiatif sebagai argumen atau parameter.
Oracle	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi DISTINCT klausa dalam SELECT pernyataan.
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi tabel di mana kendala kunci utama memiliki nama yang sama dengan tabel.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi RAISE_APPLICATION_ERROR prosedur dengan parameter ketiga.
Oracle	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana aturan migrasi tidak secara otomatis mengubah tipe NUMERIC data ke INTEGER tempat yang berlaku.
Oracle DW	Amazon Redshift	Menerapkan dukungan CONNECT BY klausa Amazon Redshift asli dalam kode yang dikonversi.
Oracle DW	Amazon Redshift	Meningkatkan migrasi data dengan secara otomatis menambahkan subtugas untuk setiap tabel atau partisi dalam lingkup migrasi. Pendekatan ini mencegah kehilangan data untuk data yang dimasukkan setelah partisi.
Teradata	Amazon Redshift	Menerapkan konversi pandangan rekursif.
Teradata	Amazon Redshift	Meningkatkan konversi prosedur tersimpan yang menggunakan BTET dan mode ANSI transaksi dengan menambahkan dukungan mode AUTOCOMMIT transaksi Amazon Redshift asli.
Teradata	Amazon Redshift	Memperbaiki konversi prosedur tersimpan yang menggunakan semantik TERADATA transaksi dengan menambahkan NONATOMIC kata kunci dalam kode yang dikonversi.
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah di mana kode yang dikonversi menyertakan ID kunci AWS akses dan kunci akses rahasia.

Catatan rilis untuk AWS SCT Build 671

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Memperbaiki kesalahan di mana AWS SCT tidak memiliki izin untuk menyimpan file proyek di Windows.
Semua	Semua	<p>Memperbarui template antarmuka baris AWS SCT perintah (CLI) berikut.</p> <ul style="list-style-type: none"> • BTEQ ScriptConversion • ConversionApply • HadoopMigration • HadoopResumeMigration • Informatika <p>Untuk informasi selengkapnya tentang template AWS SCT CLI, lihat Mendapatkan skenario CLI</p>
Semua	Amazon Redshift	Memperbaiki kesalahan saat AWS SCT tidak membuat paket ekstensi di antarmuka baris perintah (CLI).
Semua	Amazon Redshift	Menyelesaikan masalah di mana agen ekstraksi AWS SCT data tidak menggunakan AWS Snowball konfigurasi di antarmuka baris perintah (CLI).
Apache Oozie	AWS Step Functions	Menerapkan dukungan untuk migrasi dari Apache Oozie ke AWS Step Functions dalam mode antarmuka baris perintah (CLI). Setelah memigrasikan beban kerja Hadoop Anda ke Amazon EMR, Anda sekarang dapat memigrasikan sistem penjadwalan alur kerja ke AWS Cloud Untuk informasi selengkapnya, lihat Mengkonversi Apache Oozie ke AWS Step Functions .
Database Azure SQL	Aurora PostgreSQL	Memperbaiki kesalahan resolver yang terjadi untuk tabel dan alias.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	PostgreSQL	
Database Azure SQL	Aurora PostgreSQL	Menerapkan konversi INDEX ON klausa.
Microsoft SQL Server	PostgreSQL	
Database Azure SQL	Aurora PostgreSQL	Meningkatkan konversi objek berikut untuk menghindari item tindakan yang tidak terduga.
Microsoft SQL Server	PostgreSQL	<ul style="list-style-type: none"> • Pernyataan Batch • Daftar ekspresi • Alias tabel • Tabel sementara • Pemicu • Variabel pengguna
Database Azure SQL	Aurora PostgreSQL	Mengatasi kesalahan penguraian yang terjadi untuk prosedur.
Microsoft SQL Server	PostgreSQL	
Database Azure SQL	Aurora PostgreSQL	Memperbaiki kesalahan di mana AWS SCT digunakan nama tabel sementara yang salah dalam kode yang dikonversi untuk OBJECT_ID fungsi.
Microsoft SQL Server	PostgreSQL	

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Database Azure SQL Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Masalah yang diselesaikan di mana item tindakan 9996 tiba-tiba muncul selama konversi elemen kode berikut. <ul style="list-style-type: none"> • Fungsi CONVERT • Fungsi DATEADD • DELETEpernyataan di dalam fungsi inline • Pernyataan IF • INSERTatau UPDATE tindakan pada kolom • Pernyataan RETURN • UPDATEpernyataan dengan kueri atau fungsi yang kompleks
BigQuery	Amazon Redshift	Menambahkan dukungan untuk BigQuery sebagai sumber untuk proses penilaian multiserver. Untuk informasi selengkapnya, lihat Membuat laporan penilaian penilaian penilaian penilaian multiserver .
Hadoop	Amazon EMR	Memperbarui versi driver Apache Hive JDBC yang didukung yang Anda gunakan untuk terhubung ke database sumber Anda. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan .
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan pemuat metadata sumber untuk memastikan bahwa AWS SCT memuat objek database sumber seperti kunci utama, indeks implisit, dan sebagainya.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Memperbaiki kesalahan resolver yang terjadi untuk kolom dalam kursor implisit.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan kemampuan untuk menjaga pemformatan nama kolom, ekspresi, dan klausa dalam pernyataan DHTML dalam kode yang dikonversi.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan konversi kunci asing skema silang.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan konversi LENGTH dan VARCHAR fungsi.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan konversi LABEL ON dan DECLARE CONDITION pernyataan.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan konversi SELECT pernyataan dengan OPTIMIZE FOR klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi CREATE TABLE pernyataan dengan menambahkan nilai default untuk semua tipe data yang didukung.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi INCREMENT BY atribut.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi tabel dipartisi dengan menambahkan kemampuan untuk mengecualikan partisi tabel dari lingkup konversi.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi definisi kunci primer dengan INCLUDE kolom.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi SUBSTRING fungsi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi SET dan DECLARE HANDLER FOR pernyataan.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi tipe data variabel.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi XMLTABLE fungsi.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan aliran migrasi dengan menerapkan urutan berikut menerapkan objek dikonversi ke database target: tabel, partisi, indeks, kendala, kunci asing, dan pemicu.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi komentar di kode sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah saat item tindakan 9997 muncul secara tak terduga selama konversi alias dalam klausa. FROM
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah saat item tindakan 9997 muncul secara tak terduga selama konversi alias cursor.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Memperbaiki kesalahan di mana kode yang dikonversi mengembalikan hasil yang berbeda untuk SELECT pernyataan dengan ORDER BY klausa. Karena SQL Server dan PostgreSQL NULL memperlakukan nilai secara berbeda, kode yang dikonversi sekarang NULLS FIRST menyertakan NULLS LAST atau klausa yang memastikan bahwa kode yang dikonversi mengembalikan hasil dalam urutan yang sama dengan kode sumber Anda.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah di mana tipe data dalam fungsi tabel salah dikonversi.
MySQL	Amazon RDS for MySQL	Menyelesaikan masalah di mana tanda kutip tunggal (' ') tiba-tiba muncul di sekitar nama objek database dalam kode yang dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL L PostgreSQL L	Menambahkan tampilan baru ke paket ekstensi untuk meniru tampilan sistem Oracle yang Anda gunakan untuk menampilkan informasi tentang partisi dan subpartisi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Diperbarui dua fungsi dalam paket ekstensi untuk menambahkan nama skema sebagai argumen dalam kode yang dikonversi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Memperbaiki kesalahan yang AWS SCT tidak menggunakan parameter yang benar untuk konversi aplikasi C++ setelah menyegarkan kode aplikasi di antarmuka pengguna.
Oracle	Aurora PostgreSQL L PostgreSQL L	Memperbaiki konversi CREATE TYPE pernyataan untuk menghindari pengecualian yang tidak terduga.
Oracle	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi tabel bersarang.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL L PostgreSQL L	Mengatasi kesalahan penguraian yang terjadi untuk objek paket.
Oracle	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah di mana nama objek dipangkas AWS SCT secara tak terduga dalam kode yang dikonversi ketika panjang nama melebihi 60 karakter.
Oracle	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah di mana pemacu tingkat baris untuk tabel yang dipartisi salah dikonversi.
Oracle DW	Amazon Redshift	Menerapkan dukungan partisi tabel otomatis untuk migrasi data. Untuk mempercepat migrasi data, secara otomatis AWS SCT dapat partisi tabel besar atau partisi berdasarkan nilai-nilai dalam ROWID pseudocolumn. Untuk informasi selengkapnya, lihat Menggunakan partisi asli .
Teradata	Amazon Redshift	Menerapkan dukungan MERGE perintah asli dalam kode Amazon Redshift yang dikonversi. Untuk informasi selengkapnya tentang MERGE perintah di Amazon Redshift, lihat MENGGABUNGAN di Panduan Pengembang Database Amazon Redshift.
Teradata	Amazon Redshift	Memperbaiki konversi DELETE dan UPDATE pernyataan yang tidak menggunakan nama tabel eksplisit.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana IN dan NOT IN pernyataan salah dikonversi.

Catatan rilis untuk AWS SCT Build 670

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Database Azure SQL Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	<p>Masalah yang diselesaikan di mana item tindakan 9996 tiba-tiba muncul selama konversi elemen kode berikut.</p> <ul style="list-style-type: none"> • CREATE INDEX pernyataan di dalam INCLUDE pernyataan • Pernyataan DECLARE • Pernyataan DECLARE . . . TABLE • DECLARE dengan nilai default di dalam LOOP pernyataan • Pernyataan DELETE • DROP CONSTRAINT pernyataan di dalam ALTER TABLE pernyataan • EXECUTE AS CALLER dan REVERT • Pernyataan IIF • Daftar ekspresi • Fungsi MONTH() • Pernyataan UPDATE • Fungsi YEAR()
Analisis Sinaps Azure	Amazon Redshift	Menambahkan dukungan untuk Azure Synapse Analytics sebagai sumber untuk proses penilaian multiserver. Untuk informasi selengkapnya, lihat Membuat laporan penilaian penilaian penilaian penilaian penilaian multiserver .
Hadoop	Amazon EMR	Menerapkan dukungan untuk migrasi cluster Hadoop ke Amazon EMR dalam mode antarmuka baris perintah (CLI). Untuk informasi selengkapnya, lihat Memigrasi kerangka kerja data besar .
IBM Db2 untuk z/OS	Aurora PostgreSQL L	Memperbaiki kesalahan resolver yang terjadi untuk tabel sumber dan kolom.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
	PostgreSQL	
IBM Db2 untuk z/OS	Aurora PostgreSQL	Menerapkan konversi CASE ekspresi.
	PostgreSQL	
IBM Db2 untuk z/OS	Aurora PostgreSQL	Menerapkan konversi CURRENT_DATE referensi ke register khusus. Referensi ke register khusus di Db2 untuk z/OS adalah referensi ke nilai yang disediakan oleh server saat ini.
	PostgreSQL	
IBM Db2 untuk z/OS	Aurora PostgreSQL	Menerapkan konversi DATE dan POSSTR fungsi.
	PostgreSQL	
IBM Db2 untuk z/OS	Aurora PostgreSQL	Meningkatkan konversi konstanta datetime.
	PostgreSQL	
IBM Db2 untuk z/OS	Aurora PostgreSQL	Meningkatkan konversi nilai default untuk kolom tipe data berikut: DATE,, TIMESTAMPTZ , danTIMESTAMP WITH TIME ZONE.
	PostgreSQL	

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana item tindakan 9996 muncul secara tak terduga selama konversi pernyataan. <code>SELECT INTO</code>
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi <code>DATEDIFF</code> fungsi.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Memperbaiki kesalahan di mana <code>ISNULL</code> fungsi dikonversi ke <code>NULLIF</code> . Akibatnya, kode yang dikonversi menghasilkan hasil yang berbeda dibandingkan dengan kode sumber. Sekarang, AWS SCT mengkonversi <code>ISNULL</code> fungsi ke <code>COALESCE</code> .
Netezza	Amazon Redshift	Agen ekstraksi data yang ditingkatkan untuk menyelesaikan masalah di mana status gagal ditetapkan untuk tugas yang berhasil diselesaikan.
Netezza	Amazon Redshift	Menambahkan kemampuan untuk mengubah titik akhir dalam subtugas setelah memulai migrasi data dengan agen ekstraksi data.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	Aurora MySQL	Ditambahkan kemampuan untuk terhubung ke database menggunakan protokol alamat IPv6.
MySQL	Aurora PostgreSQL	
Oracle	MySQL	
PostgreSQL	PostgreSQL	
Oracle	Amazon RDS for Oracle	Konversi DBMS_JOB paket yang diimplementasikan yang menjadwalkan dan mengelola pekerjaan dalam antrian pekerjaan.
Oracle	Aurora PostgreSQL	Menambahkan fungsi baru ke paket ekstensi untuk meningkatkan konversi tabel bersarang global. Fungsi-fungsi baru ini meniru DELETE, EXTEND, dan TRIM berfungsi dalam kode Oracle sumber Anda.
Oracle	Aurora PostgreSQL	Ditambahkan kemampuan untuk menentukan lingkup konversi untuk kode SQL yang tertanam dalam aplikasi Java. Anda sekarang dapat mengecualikan himpunan bagian dari proyek aplikasi sumber dari lingkup konversi. Untuk informasi selengkapnya, lihat Mengonversi kode SQL aplikasi Java Anda AWS SCT .
Oracle	Aurora PostgreSQL	Peningkatan konversi operator penggabungan () di dalam indeks fungsional.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi IN kondisi di mana kode sumber Anda tidak menyertakan tanda kurung untuk satu ekspresi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi MERGE pernyataan ke INSERT ON CONFLICT dalam PostgreSQL.
Oracle	Aurora PostgreSQL L PostgreSQL L	Mengatasi kesalahan penguraian yang terjadi untuk paket prosedur.
Oracle	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah saat item tindakan 5072 muncul secara tak terduga selama konversi paket.
Oracle DW	Amazon Redshift	Memperbaiki kesalahan yang AWS SCT tidak menerapkan paket ekstensi saat menerapkan kode yang dikonversi ke database target.
Oracle DW	Amazon Redshift	Memperbaiki kesalahan yang AWS SCT tidak menerapkan beberapa file paket ekstensi saat menggunakan wizard paket ekstensi.
Oracle DW	Amazon Redshift	Menyelesaikan masalah yang AWS SCT tidak dapat memproses migrasi data AWS Snowball dengan lebih dari 500 tugas yang berjalan secara paralel.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle DW	Amazon Redshift	Menyelesaikan masalah di mana fungsi yang ditentukan pengguna dengan tipe yang ditentukan pengguna salah dikonversi.

Catatan rilis untuk AWS SCT Build 669

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Meningkatkan proses penilaian multiserver, yang membantu menentukan platform basis data target yang optimal untuk basis data sumber Anda. Sekarang, AWS SCT abaikan AWS Secrets Manager kunci jika Anda memberikan kredensial database dalam file input comma separated values (CSV). Untuk informasi selengkapnya, lihat Membuat laporan penilaian penilaian penilaian penilaian penilaian multiserver .
Semua	Semua	Menyelesaikan masalah di mana laporan penilaian multiserver menyertakan alamat IP database sumber Anda saat menggunakan rahasia dari AWS Secrets Manager untuk terhubung ke database.
Semua	Amazon Redshift	Menerapkan konfigurasi otomatis pengaturan mesin virtual Java (JVM) tergantung pada sistem operasi dan RAM yang tersedia. AWS SCT menggunakan JVM ini untuk menjalankan pekerjaan agen ekstraksi data.
Semua	Amazon Redshift	Menyelesaikan masalah di mana agen ekstraksi data tidak dimulai di Ubuntu.
Semua	Amazon Redshift	Menyelesaikan masalah di mana tugas ekstraksi data tidak dimulai setelah menjalankan <code>StartAgent.bat</code> file di Windows.
Database Azure SQL	Aurora PostgreSQL	Menyelesaikan masalah di mana nama kolom salah dikonversi dengan opsi Hasilkan nama unik untuk indeks diaktifkan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	PostgreSQL	
Greenplum	Amazon Redshift	Menerapkan konversi fungsi yang kembali VOID ke prosedur.
Greenplum	Amazon Redshift	Menyelesaikan masalah saat migrasi data gagal saat database sumber tidak menyertakan nilai numerik (NaN) dalam kolom numerik. AWS SCT agen ekstraksi data sekarang mengganti nilai NaN dengan NULL.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menambahkan pengaturan konversi baru untuk menentukan DATE FORMAT dan TIME FORMAT opsi selama konversi fungsi CHAR bawaan.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menambahkan item tindakan 8534 untuk konversi kursor yang telah ditentukan yang dideklarasikan dengan klausa. WITHOUT RETURN Jika kursor Anda tidak mengembalikan kumpulan hasil, AWS SCT berikan NULL nilai ke nama kursor Anda dalam kode yang dikonversi dan memunculkan item tindakan.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Mengedit CURRENT_CLIENT_APPLNAME properti yang mengidentifikasi AWS SCT selama koneksi ke database sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan pengaturan konversi baru untuk menentukan DATE FORMAT dan TIME FORMAT opsi selama konversi fungsi CHAR bawaan.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Konversi yang diimplementasikan dari LEAVE pernyataan dalam pernyataan BEGIN . . . END blok.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan konversiXMLPARSE,XMLTABLE, dan XMLNAMESPACES fungsi.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Peningkatan konversi fungsi CHAR bawaan.
IBM Db2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Peningkatan konversi kursor.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi pernyataan loop. FOR
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi dari jenis tabel penggunaan dalam SELECT pernyataan.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL L	Menerapkan dukungan versi baru 2.2.0 dari file konfigurasi fitur Babelfish. File ini mendefinisikan fitur SQL yang didukung dan tidak didukung oleh versi Babelfish tertentu.
Netezza	Amazon Redshift	Agen ekstraksi data yang ditingkatkan untuk menyelesaikan masalah di mana satu baris tidak dihapus dari tabel target selama replikasi data yang sedang berlangsung.
Oracle	Amazon RDS for Oracle	Peningkatan konversi fitur Oracle Database Enterprise Edition.
Oracle	Aurora PostgreSQL L PostgreSQL L	Konversi GROUPING_ID fungsi yang diimplementasikan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi kode SQL dalam aplikasi C # dengan menambahkan dukungan pemetaan tipe data kustom dalam mode antarmuka baris perintah (CLI).
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi tabel bersarang untuk menghindari item tindakan yang tidak terduga 9996.
Oracle	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah di mana panggilan konstruktor objek salah dikonversi.
Oracle DW	Amazon Redshift	Menerapkan dukungan partisi tabel yang ada untuk migrasi data. Untuk mempercepat migrasi data, AWS SCT buat subtugas untuk setiap partisi tabel sumber yang tidak kosong. Untuk informasi selengkapnya, lihat Menggunakan partisi asli .
Teradata	Amazon Redshift	Peningkatan konversi CAST fungsi dengan <code>TIME WITH TIME ZONE AS TIMESTAMP</code> , <code>TIME WITH TIME ZONE AS CHAR</code> , dan <code>TIMESTAMP AS TIME WITH TIME ZONE</code> argumen.
Teradata	Amazon Redshift	Peningkatan konversi CAST fungsi dengan <code>FORMAT</code> opsi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana <code>CEIL</code> fungsi tidak dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Menyelesaikan masalah di mana MERGE pernyataan dengan DELETE klausa salah dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana TO_CHAR fungsi dengan argumen tanggal dan format salah dikonversi.

Catatan rilis untuk AWS SCT Build 668

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Amazon Redshift	Menyelesaikan masalah di mana operator perkalian dalam aturan migrasi tidak berfungsi dengan benar. Operator ini memungkinkan untuk mengubah panjangchar,, varcharnvarchar, dan tipe string data. Untuk informasi selengkapnya, lihat Membuat aturan migrasi .
Analisis Sinaps Azure	Amazon Redshift	Menerapkan dukungan CONVERT fungsi dengan VARCHAR argumen.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi SELECT pernyataan dengan NOLOCK klausa.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi UPDATE pernyataan dengan alias atau dengan SET dan FROM klausa.
Greenplum	Amazon Redshift	Menerapkan partisi virtual otomatis untuk migrasi data. AWS SCT menggunakan kolom GP_SEGMENT_ID sistem untuk membuat partisi.
Greenplum	Amazon Redshift	Menerapkan dukungan RETURN QUERY dan RETURN SETOF klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Greenplum	Amazon Redshift	Menerapkan dukungan SUBSTRING fungsi dengan tiga parameter.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi SUBSTR fungsi dengan LOCATE parameter.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Ditambahkan pilihan untuk menggunakan array REFCURSOR variabel untuk mengembalikan set hasil dinamis. Saat Anda memilih opsi ini di pengaturan konversi, AWS SCT tambahkan OUT parameter tambahan dalam kode yang dikonversi.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Dukungan yang diimplementasikan dari pernyataan FOR loop.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menerapkan dukungan XMLPARSE fungsi. Menambahkan item tindakan 8541 untuk striping spasi putih dalam XMLPARSE fungsi.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Meningkatkan konversi beberapa penanganan pengecualian dalam satu BEGIN . . . END blok.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	Aurora PostgreSQL	Peningkatan konversi INSERT dan DELETE pemicu.
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	Peningkatan konversi panggilan prosedur bersarang.
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	Peningkatan konversi jenis tabel.
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	Menyelesaikan masalah di mana NOT operasi logis bitwise salah dikonversi untuk nilai integer.
	PostgreSQL	
Microsoft SQL Server	Aurora PostgreSQL	Menyelesaikan masalah di mana array lokal tidak diinisialisasi di PostgreSQL versi 8.0.2 dan yang lebih rendah.
	PostgreSQL	

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana MERGE pernyataan dengan WHEN NOT MATCHED BY SOURCE klausa salah dikonversi.
MySQL	Aurora MySQL	Menyelesaikan masalah ketika AWS SCT salah menentukan izin pengguna yang diberikan oleh <code>rds_superuser_role</code> peran.
Netezza	Amazon Redshift	Meningkatkan pemuat metadata sumber untuk memastikan bahwa memuat objek database AWS SCT dengan benar dengan nama dalam huruf kecil.
Oracle	Aurora PostgreSQL PostgreSQL	Menambahkan fungsi baru ke paket ekstensi untuk meningkatkan konversi tabel bersarang lokal. Fungsi-fungsi baru ini meniru <code>PRIOR</code> , <code>NEXT</code> , <code>LIMIT</code> , <code>FIRST</code> , <code>LAST</code> , <code>EXISTS</code> , <code>EXTEND</code> , <code>TRIM</code> , <code>DELETE</code> , dan <code>SET</code> fungsi dalam kode Oracle sumber Anda. Untuk informasi selengkapnya, lihat Menggunakan paket ekstensi .
Oracle	Aurora PostgreSQL PostgreSQL	Ditambahkan kemampuan untuk menentukan lingkup konversi untuk aplikasi C#. Pengguna sekarang dapat mengecualikan himpunan bagian dari proyek aplikasi sumber dari lingkup konversi.
Oracle	Aurora PostgreSQL PostgreSQL	Menerapkan dukungan COUNT metode dalam koleksi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL L PostgreSQL L	Menerapkan dukungan variabel dan konstruktor dalam tabel bersarang.
Oracle	Aurora PostgreSQL L PostgreSQL L	Menerapkan dukungan RATIO_TO_REPORT dan STANDARD_HASH fungsi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi objek besar (LOB) sebagai bagian dari paket AWS SCT ekstensi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi koleksi lokal.
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi JOIN pernyataan dengan USING klausa di mana nama kolom tidak menyertakan nama tabel.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL L PostgreSQL L	Konversi EMPTY_BLOB dan EMPTY_CLOB fungsi yang diimplementasikan.
Oracle	Aurora PostgreSQL L PostgreSQL L	Konversi yang diimplementasikan dari variabel pengikat posisi dalam aplikasi C #.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Konversi yang diimplementasikan dari pemacu multi-peristiwa.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Konversi pemacu rekursif yang diterapkan.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi pemacu dengan variabel @@rowcount global.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SAP ASE	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana fungsi agregat dalam SET klausa UPDATE pernyataan salah dikonversi.
SAP ASE	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana item tindakan 42702 tiba-tiba muncul selama konversi pernyataan. UPDATE
SAP ASE	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana CONVERT fungsi dengan CHAR argumen salah dikonversi.
Kepingan salju	Amazon Redshift	Menambahkan dukungan Snowflake sebagai sumber migrasi data dengan agen ekstraksi AWS SCT data. Untuk informasi selengkapnya, lihat Memigrasi data dari gudang data lokal ke Amazon Redshift .
Teradata	Amazon Redshift	Peningkatan konversi CAST fungsi dengan TIMESTAMP AS TIME WITH TIMEZONE argumen.

Catatan rilis untuk AWS SCT Build 667

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menerapkan dukungan Informatica ekstrak, transformasi, dan muat (ETL) skrip dalam mode antarmuka baris perintah (CLI). AWS SCT secara otomatis mengalihkan skrip ETL Informatica Anda ke database target baru. Juga, AWS SCT mengkonversi nama objek

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		dan kode SQL yang tertanam dalam objek Informatica Anda. Untuk informasi selengkapnya, lihat Mengkonversi skrip ETL Informatica .
Semua	Amazon Redshift	Meningkatkan versi driver minimum yang didukung untuk Amazon Redshift menjadi 2.1.0.9. Untuk informasi selengkapnya, lihat Mengunduh driver database yang diperlukan .
Analisis Sinaps Azure	Amazon Redshift	Menambahkan fungsi baru ke paket ekstensi untuk meningkatkan konversi CONVERT fungsi dengan tiga argumen tanggal dan waktu.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi DATEDIFF fungsi.
Analisis Sinaps Azure	Amazon Redshift	Memperbarui versi paket ekstensi. Pastikan Anda menerapkan versi terbaru dari paket ekstensi dalam AWS SCT proyek yang ada. Untuk informasi selengkapnya, lihat Menggunakan paket ekstensi .
Microsoft SQL Server DW		
BigQuery	Amazon Redshift	Menyelesaikan masalah di mana objek yang difilter tidak dikonversi dalam mode antarmuka baris perintah (CLI).
Greenplum	Amazon Redshift	Memperbaiki kesalahan saat AWS SCT tidak mengonversi tabel sementara yang dideklarasikan dalam prosedur tersimpan.
Greenplum	Amazon Redshift	Memperbaiki kesalahan di mana atribut pengkodean kolom hilang dalam kode yang dikonversi.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Konversi UPDATE pernyataan yang diterapkan untuk tabel referensi diri yang memiliki lebih dari satu INNER JOIN klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Menerapkan dukungan <code>inserted</code> dan tabel <code>deleted</code> sementara yang SQL Server gunakan untuk pemicu DHTML.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Peningkatan konversi tipe yang ditentukan pengguna dalam prosedur tersimpan yang dibuat dalam skema database yang berbeda. Menyelesaikan masalah di mana AWS SCT tidak dapat menemukan tipe data dan menampilkan item tindakan 9996.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana tanda kurung siku (<code>[]</code>) tiba-tiba muncul di sekitar nama objek database dalam kode yang dikonversi.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana <code>@@ROWCOUNT</code> fungsi salah dikonversi.
Microsoft SQL Server DW	Amazon Redshift	Menerapkan dukungan <code>geometry</code> dan tipe <code>geography</code> data.
Microsoft SQL Server DW	Amazon Redshift	Menerapkan dukungan MAX kata kunci dalam deklarasi tipe data dalam kode yang dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server DW	Amazon Redshift	Peningkatan konversi DATEADD fungsi.
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan konversi kode SQL dalam aplikasi Java dengan menambahkan dukungan untuk MyBatis kerangka kerja. Untuk informasi selengkapnya, lihat Mengonversi kode SQL dalam aplikasi Java .
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan konversi kode SQL dalam aplikasi Java yang menggunakan MyBatis kerangka kerja. Menambahkan item tindakan 30411 untuk kode SQL dengan sintaks yang tidak didukung.
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan konversi kode SQL dalam aplikasi Pro* C dengan menambahkan dukungan untuk deklarasi. <code>typedef struct</code>
Oracle	Aurora PostgreSQL PostgreSQL	Menerapkan dukungan CROSS JOIN dan LEFT JOIN pernyataan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL	Peningkatan konversi MERGE pernyataan. Menyelesaikan masalah di mana nilai yang akan disisipkan tidak ada dalam kode yang dikonversi.
	PostgreSQL	
Teradata	Amazon Redshift	Mengubah setelan pengkodean kompresi kolom default yang AWS SCT digunakan dalam kode yang dikonversi agar sesuai dengan setelan Amazon Redshift default. Untuk informasi selengkapnya, lihat Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana operasi matematika yang menggunakan tipe TIME data salah dikonversi.
Teradata	Amazon Redshift RSQL	Menerapkan konversi FastExport kode yang ada di dalam skrip shell.
Teradata BTEQ	Amazon Redshift RSQL	Memperbaiki kesalahan di mana AWS SCT tidak dikonversi COALESCE dan %data pernyataan.
Vertica	Amazon Redshift	Saran optimasi konversi yang ditingkatkan saat pengguna memilih satu strategi pengoptimalan.

Catatan rilis untuk AWS SCT Build 666

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Database Azure SQL	Aurora PostgreSQL	Mengatasi kesalahan penguraian yang terjadi untuk ON klausa yang ada di dalam pernyataan. JOIN

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	PostgreSQL	
Analisis Sinaps Azure	Amazon Redshift	Menambahkan tiga fungsi baru ke paket ekstensi untuk meningkatkan konversi CONVERT fungsi dengan argumen tanggal dan waktu.
Analisis Sinaps Azure	Amazon Redshift	Meningkatkan pemuat metadata sumber untuk memastikan bahwa AWS SCT memuat skema database sistem.
Analisis Sinaps Azure	Amazon Redshift	Memperbaiki kesalahan resolver yang terjadi untuk kolom tabel sementara.
Analisis Sinaps Azure	Amazon Redshift	Menerapkan konversi BINARY dan tipe VARBINARY data ke tipe VARBYTE data.
Analisis Sinaps Azure	Amazon Redshift	Dukungan yang diimplementasikan dari tipe TIME data dalam kode yang dikonversi.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi COLLATE klausa. Menyelesaikan masalah saat item tindakan 31141 muncul secara tak terduga selama konversi kolom dengan pemeriksaan basis data default.
BigQuery	Amazon Redshift	Menerapkan konversi prosedur yang mengubah parameter input.
Greenplum	Amazon Redshift	Menyelesaikan masalah saat AWS SCT menggunakan kueri yang tidak kompatibel dengan database Greenplum 6.x.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi EXCEPTION bagian dengan mentransfer penanganan pengecualian dari Db2 untuk z/OS ke PostgreSQL.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi OPEN CURSOR pernyataan.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Menerapkan konversi IIF fungsi dengan menggunakan CASE ekspresi.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah di mana prosedur dengan parameter bernilai tabel salah dikonversi saat CREATE PROCEDURE pernyataan tidak menyertakan BEGIN...END blok.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah di mana SCOPE_IDENTITY fungsi tersebut salah dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Amazon RDS for Oracle	Memperbaiki kesalahan loader yang terjadi dengan SELECT_CATALOG_ROLE peran saat menggunakan Oracle 10g sebagai sumber.
Oracle	Amazon RDS for Oracle	Meningkatkan loader untuk mendukung pekerjaan Oracle Scheduler.
Oracle	Aurora PostgreSQL PostgreSQL	Menerapkan konversi JOIN pernyataan dengan USING klausa.
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan kinerja kode yang dikonversi di mana kode sumber mencakup variabel global dalam WHERE klausa.
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan konversi kode SQL dalam aplikasi Java dengan menambahkan dukungan untuk MyBatis kerangka kerja. Untuk informasi selengkapnya, lihat Mengonversi kode SQL dalam aplikasi Java .
Oracle DW	Amazon Redshift	Konversi yang diimplementasikan dari operator PIVOT dan UNPIVOT relasional.
Teradata	Amazon Redshift	Memperbaiki kesalahan di mana kode sumber yang menggunakan objek JSON tidak dikonversi.
Teradata	Amazon Redshift	Memperbaiki kesalahan di mana tabel yang dibuat oleh pengguna yang jatuh tidak dimuat dengan benar.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Menerapkan konversi INSTR fungsi ke fungsi Amazon Redshift STRPOS asli.
Teradata	Amazon Redshift	Konversi yang diimplementasikan dari NVP dan TRANSLATE fungsi.
Teradata	Amazon Redshift	Peningkatan konversi COALESCE ekspresi.
Teradata	Amazon Redshift	Peningkatan konversi DECLARE CONDITION pernyataan.
Teradata	Amazon Redshift	Peningkatan konversi EXTRACT fungsi dengan elemen SECOND sintaks.
Teradata	Amazon Redshift	Peningkatan konversi SQLSTATE dan SQLCODE variabel di dalam LOOP pernyataan.
Teradata	Amazon Redshift	Peningkatan konversi indeks unik.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi CURRENT_TIMESTAMP pernyataan dengan presisi fraksional disetel ke 3.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana garis miring terbalik salah dikonversi dalam literal string.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana EXEC pernyataan yang dikonversi menyertakan nama bidang yang salah dalam ADD CONSTRAINT pernyataan.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana QUALIFY subquery dikonversi termasuk nama subquery yang salah.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Menyelesaikan masalah saat tampilan yang dikonversi tidak diterapkan. Menambahkan cast eksplisit ke tipe data tertentu untuk nilai NULL dalam kode yang dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana fungsi tanggal dan waktu salah dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana literal string heksadesimal tidak dikonversi.

Catatan rilis untuk AWS SCT Build 665

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Analisis Sinaps Azure	Amazon Redshift	Konversi CONCAT fungsi yang diimplementasikan dengan VARCHAR argumen.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi CREATE TABLE pernyataan yang membuat tabel sementara dan tidak menyertakan nama skema. AWS SCT membuat dbo skema untuk menyimpan tabel sementara ini dalam database target.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi DROP TABLE pernyataan yang Anda jalankan pada tabel sementara.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi OBJECT_ID pernyataan dengan BEGIN...END blok.
Analisis Sinaps Azure	Amazon Redshift	Menyelesaikan kesalahan di mana AWS SCT tidak dapat mengonversi prosedur yang disimpan dengan komentar blok.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
BigQuery	Amazon Redshift	Menerapkan konversi gudang BigQuery data ke Amazon Redshift. Untuk informasi selengkapnya, lihat Menggunakan BigQuery sebagai sumber untuk AWS SCT .
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Peningkatan konversi pemicu yang menangani beberapa peristiwa dan bekerja dengan <code>inserted</code> dan tabel <code>deleted</code> sistem di SQL Server.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Memperbaiki kesalahan resolver yang terjadi untuk <code>inserted</code> dan tabel <code>deleted</code> sistem di SQL Server.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Menerapkan dukungan versi baru 2.1.0 dari file konfigurasi fitur Babelfish. File ini mendefinisikan fitur SQL yang didukung dan tidak didukung oleh versi Babelfish tertentu.
Oracle	Aurora MySQL MariaDB MySQL	Menyelesaikan masalah di mana tipe <code>varchar2</code> data salah dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora MySQL	Untuk database Oracle versi 12c dan yang lebih tinggi, AWS SCT mendukung tipe data tambahan berikut:
	Aurora PostgreSQL	<ul style="list-style-type: none"> • VARCHAR2 • NVARCHAR2 • RAW
	MariaDB	
	MySQL	AWS SCT meningkatkan panjang kolom maksimum yang didukung dari 8.000 menjadi 32.767 byte untuk tipe data ini.
	PostgreSQL	
	L	
Oracle	Aurora PostgreSQL	Mengatasi kesalahan parsing yang terjadi untuk paket Oracle Event Processing.
	L	
	PostgreSQL	
	L	
Teradata	Amazon Redshift	Menambahkan item tindakan 13214 untuk beberapa RESET WHEN klausa dalam satu pernyataan. SELECT
Teradata	Amazon Redshift	Menambahkan item tindakan untuk SQLSTATE variabel yang terletak di luar blok penanganan pengecualian.
Teradata	Amazon Redshift	Konversi ACTIVITY_COUNT variabel yang diterapkan keROW_COUNT .
Teradata	Amazon Redshift	Konversi yang diterapkan dari ST_TRANSFORM fungsi geometri bawaan.
Teradata	Amazon Redshift	Peningkatan konversi pernyataan hapus dalam tampilan tanpa WHERE klausa.
Teradata	Amazon Redshift	Peningkatan konversi CAST operator dalam ekspresi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Peningkatan konversi GROUP BY klausa.
Teradata	Amazon Redshift	Peningkatan konversi INSTR dan fungsi REGEXP_INSTR bawaan.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana referensi alias kolom lateral salah dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana nama kolom salah dikonversi dalam QUALIFY subquery.
Teradata	Amazon Redshift	Menerapkan konversi .QUIT perintah dengan kata kunci nilai ERRORCODE status.
Teradata BTEQ	Amazon Redshift RSQL	Menyelesaikan masalah di mana item tindakan 9996 muncul secara tak terduga selama konversi pernyataan. CREATE
Teradata BTEQ	Amazon Redshift RSQL	Menyelesaikan masalah di mana item tindakan 9998 muncul secara tak terduga selama konversi pernyataan. END

Catatan rilis untuk AWS SCT Build 664

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan dukungan Amazon Redshift Serverless sebagai sumber dan target untuk proyek migrasi database di AWS SCT Untuk terhubung ke Amazon Redshift Tanpa Server, pastikan Anda menggunakan driver Amazon Redshift JDBC versi 2.1.0.9 atau lebih tinggi.
Semua	Semua	Meningkatkan antarmuka pengguna jendela Pengaturan konversi. AWS SCT sekarang menampilkan pengaturan hanya untuk

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	
		pasangan konversi database dengan aturan pemetaan yang dibuat. Untuk informasi selengkapnya, lihat Membuat aturan pemetaan .	
Semua	Semua	Memperbarui laporan penilaian untuk menghapus informasi duplikat tentang garis dan posisi item tindakan.	
Semua	Amazon Redshift	Menerapkan penyeimbangan memori otomatis dalam tugas ekstraksi data.	
Semua	Amazon Redshift	Mengatasi kesalahan di mana agen ekstraksi data tidak dapat terhubung ke AWS Snowball perangkat.	
Database Azure SQL	Aurora MySQL	Menerapkan dukungan SUSE Linux 15.3 sebagai platform untuk menjalankan agen ekstraksi data.	
IBM Db2 untuk z/OS	Aurora PostgreSQL		
IBM Db2 LUW	MariaDB		
Microsoft SQL Server	MySQL PostgreSQL		
MySQL			
Oracle			
PostgreSQL			
SAP ASE			
Analisis Sinaps Azure	Amazon Redshift		Peningkatan konversi DATEADD fungsi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menambahkan kemampuan untuk mengubah pemeriksaan kolom dalam aturan migrasi.
Microsoft SSIS	AWS Glue AWS Glue Studio	Mengatasi kesalahan tak terduga yang terjadi saat pengguna memilih skrip sumber.
Oracle	Aurora MySQL MariaDB MySQL	Konversi yang diimplementasikan dari penggunaan fungsi yang disimpan sebagai ekspresi kolom yang dihasilkan. AWS SCT menciptakan pemacu untuk meniru perilaku ini karena MySQL tidak mendukung penggunaan fungsi tersimpan sebagai ekspresi kolom yang dihasilkan.
Oracle	Aurora PostgreSQL L PostgreSQL L	Menerapkan konversi fungsi dari UTL_MATCH paket sebagai bagian dari paket AWS SCT ekstensi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Konversi REGEXP_LIKE fungsi yang diimplementasikan dengan NULL parameter.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL	Peningkatan konversi SYS_EXTRACT_UTC fungsi.
	L PostgreSQL L	
Oracle	Aurora PostgreSQL	Peningkatan konversi kode SQL dalam aplikasi C ++ dengan menerapkan dukungan Wcscats, Wcscpys, dan Wcsncats fungsi. Untuk informasi selengkapnya, lihat Mengonversi kode SQL dalam aplikasi C ++ dengan AWS SCT .
	L PostgreSQL L	
Oracle DW Kepingan salju	Amazon Redshift	Menyelesaikan masalah di mana pernyataan yang dikonversi tidak menyertakan konversi nilai eksplisit ke tipe data kolom. Masalah ini terjadi dalam pernyataan yang menggunakan hasil kueri dari tabel lain.
Teradata	Amazon Redshift	Menambahkan kemampuan untuk mengubah pemeriksaan kolom antara case sensitive dan case insensitive dalam aturan migrasi. Untuk informasi selengkapnya, lihat Membuat aturan migrasi .
Teradata	Amazon Redshift	Memperbaiki kesalahan resolver yang terjadi untuk CREATE TABLE AS pernyataan.
Teradata	Amazon Redshift	Memperbaiki kesalahan di mana P_INTERSECT fungsi bawaan dengan COALESCE ekspresi tidak dikonversi.
Teradata	Amazon Redshift	Menerapkan konversi kolom bernama OID _OID untuk menghindari penggunaan kata kunci yang dicadangkan di Amazon Redshift.
Teradata	Amazon Redshift	Menerapkan konversi RENAME pernyataan untuk fungsi, prosedur, tampilan, dan makro.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Menerapkan konversi <code>STROKE</code> fungsi ke <code>SPLIT_PART</code> fungsi di Amazon Redshift.
Teradata	Amazon Redshift	Peningkatan konversi fungsi <code>INSTR</code> dan <code>REGEXP_INSTR</code> sistem.
Teradata	Amazon Redshift	Peningkatan konversi tipe <code>TIME</code> data.
Teradata	Amazon Redshift	Peningkatan emulasi <code>SET</code> dan <code>MULTISET</code> tabel dengan menerapkan konversi indeks unik primer dan sekunder.
Teradata	Amazon Redshift	Mengatasi kesalahan penguraian yang terjadi untuk <code>CHARACTER</code> fungsi tersebut.
Teradata BTEQ	Amazon Redshift RSQL	Mengatasi kesalahan yang terjadi saat pengguna menghapus skrip Teradata Basic Teradata Query (BTEQ) dari proyek. AWS SCT

Catatan rilis untuk AWS SCT Build 663

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan kemampuan untuk mengubah <code>panjangchar</code> , <code>varcharnvarchar</code> , dan tipe <code>string</code> data menggunakan operator perkalian dalam aturan migrasi. Untuk informasi selengkapnya, lihat Membuat aturan migrasi .
Semua	Semua	Menerapkan dukungan tiga kolom baru dalam laporan penilaian multiserver dan memperbarui format file input. Pastikan Anda menggunakan template yang diperbarui dari file input dengan versi terbaru AWS SCT. Untuk informasi selengkapnya, lihat Membuat laporan penilaian multiserver untuk migrasi database .

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi OBJECT_ID pernyataan.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Menambahkan dukungan untuk Babelfish untuk Aurora PostgreSQL 1.2.0 sebagai platform target untuk laporan penilaian migrasi database. Untuk informasi selengkapnya, lihat Fungsionalitas yang didukung di Babelfish menurut versi di Panduan Pengguna Amazon Aurora.
Microsoft SQL Server DW	Amazon Redshift	Menambahkan dukungan untuk AT TIME ZONE klausa.
Microsoft SQL Server DW	Amazon Redshift	Menyelesaikan masalah di mana pernyataan di luar BEGIN/END blok salah dikonversi.
Netezza	Amazon Redshift	Peningkatan konversi tipe TIME data dan konversi yang diimplementasikan dari fungsi, ekspresi, dan literal bawaan terkait.
Oracle	Aurora PostgreSQL PostgreSQL	Memperbaiki kesalahan loader yang terjadi saat menggunakan Oracle 10g sebagai sumber.
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan konversi OFFSET dan FETCH klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah di mana prosedur dengan OUT parameter dengan nilai default salah dikonversi.
Oracle DW	Amazon Redshift	Peningkatan konversi fungsi Oracle ke fungsi yang ditentukan pengguna Amazon Redshift.
Kepingan salju	Amazon Redshift	Peningkatan konversi WITH klausa.
Teradata	Amazon Redshift	Menambahkan item tindakan baru 13209 untuk karakter multibyte yang tidak didukung untuk tipe data. CHAR
Teradata	Amazon Redshift	Memperbaiki kesalahan pemuat di mana tabel tidak terisi penuh.
Teradata	Amazon Redshift	Memperbaiki kesalahan transformator di mana P_INTERSECT fungsi bawaan dalam suatu JOIN kondisi tidak dikonversi.
Teradata	Amazon Redshift	Memperbaiki masalah di mana nama tampilan diubah dalam kasus yang salah ketika SELECT pernyataan dijalankan di atas meja dengan karakter khusus dalam namanya.
Teradata	Amazon Redshift	Peningkatan konversi INSERT pernyataan dengan UNTIL_CHANGED nilai dalam tipe PERIOD(DATE) data.
Teradata	Amazon Redshift	Peningkatan konversi fungsi bawaan menggunakan FORMAT TO_CHAR fungsi di Amazon Redshift.
Teradata	Amazon Redshift	Peningkatan konversi RANK fungsi bawaan untuk memastikan bahwa kode yang dikonversi mengembalikan nilai NULL dalam urutan yang sama dengan kode sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Peningkatan konversi kendala unik seperti indeks unik primer atau sekunder.

Catatan rilis untuk AWS SCT Build 662

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan kemampuan untuk secara otomatis membuat AWS SCT proyek untuk setiap database sumber saat membuat laporan penilaian multiserver. Dengan opsi ini diaktifkan, AWS SCT dapat menambahkan aturan pemetaan untuk proyek-proyek ini dan menyimpan statistik konversi untuk penggunaan offline. Untuk informasi selengkapnya, lihat Membuat laporan penilaian multiserver untuk migrasi database .
Semua	Semua	Menerapkan dukungan persen (%) sebagai wildcard dalam database dan nama skema saat membuat laporan penilaian multiserver.
Semua	Aurora MySQL Aurora PostgreSQL	Memperbarui runtime semua AWS Lambda fungsi ke Python versi 3.9.
Semua	Amazon Redshift	Upgrade semua agen ekstraksi data untuk digunakan AWS SDK for Java 2.x.
Database Azure SQL Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Peningkatan konversi DELETE pernyataan dengan NON EXISTS klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Analisis Sinaps Azure	Amazon Redshift	Menyelesaikan kesalahan di mana koneksi ke database sumber gagal.
IBM Db2 untuk z/OS	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan kesalahan di mana kode dikonversi dari pemacu termasuk dua penyebutan alias objek.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi objek dengan nama dalam kasus campuran ketika Treat nama objek database sebagai opsi case sensitive dihidupkan.
Microsoft SQL Server DW Teradata	Amazon Redshift	Konversi yang diimplementasikan dari operator PIVOT dan UNPIVOT relasional.
Netezza	Amazon Redshift	Konversi yang diimplementasikan dari tipe TIME data.
Oracle	Aurora MySQL Aurora PostgreSQL L MySQL PostgreSQL L	Menerapkan UTL_TCP.CRLF paket konversi konstan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL L PostgreSQL L	Memperbaiki masalah paket ekstensi di mana panjang tipe data untuk kolom dengan panjang variabel tidak dipertahankan selama konversi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Menerapkan konversi kode SQL dalam aplikasi C ++. Untuk informasi selengkapnya, lihat Mengonversi kode SQL dalam aplikasi C ++ dengan AWS SCT .
Oracle	Aurora PostgreSQL L PostgreSQL L	Menerapkan dukungan penamaan case sensitive untuk konversi variabel global dan array asosiatif.
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi TO_CHAR, TO_DATE, dan TO_NUMBER fungsi dalam paket ekstensi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi TABLE () operator.
Oracle DW	Amazon Redshift	Ditambahkan dukungan untuk konversi kunci utama dan kendala lainnya.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle DW	Amazon Redshift	Memperbaiki masalah saat item tindakan 12054 tidak muncul selama konversi pernyataan bersyarat.
SAP ASE	Aurora PostgreSQL	Menyelesaikan kesalahan saat objek dengan nama kosong dibuat di pohon target selama konversi tabel dengan kolom tipe yang ditentukan pengguna.
SAP ASE	Aurora PostgreSQL	Memperbaiki kesalahan loader untuk objek yang disimpan seperti skrip, rutinitas, dan sebagainya.
Kepingan salju	Amazon Redshift	Memperbaiki masalah saat item tindakan 22152 tidak muncul saat diperlukan dan AWS SCT menampilkan hasil konversi sebagai komentar.
Kepingan salju	Amazon Redshift	Peningkatan konversi fungsi tanggal dan waktu; menerapkan dukungan zona waktu.
Kepingan salju	Amazon Redshift	Menyelesaikan masalah di mana ekspresi tabel umum non-rekursif (CTE) dengan WITH klausa dikonversi sebagai CTE rekursif.
Teradata	Amazon Redshift	Peningkatan konversi UPDATE pernyataan dengan tautan tabel dalam kondisi.
Teradata	Amazon Redshift	Peningkatan konversi RENAME TABLE pernyataan.
Teradata	Amazon Redshift	Menyelesaikan masalah saat kolom kosong muncul di file nilai dipisahkan koma (CSV) dengan laporan penilaian.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift RSQL	Memperbaiki kesalahan saat titik koma hilang di akhir makro Basic Teradata Query (BTEQ) yang dikonversi.
Teradata	Amazon Redshift RSQL	Peningkatan konversi beberapa nilai tipe data dalam CASE pernyataan.
Teradata	Amazon Redshift RSQL	Peningkatan konversi LIKE ANY klausa dengan ESCAPE karakter.
Teradata	Amazon Redshift RSQL	Peningkatan konversi CAST fungsi dalam INSERT pernyataan.
Teradata	Amazon Redshift RSQL	Peningkatan konversi zona waktu, menerapkan pemetaan wilayah zona waktu.
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah di mana item tindakan 9998 tiba-tiba muncul selama konversi skrip shell dengan skrip BTEQ.
Teradata	Amazon Redshift RSQL AWS Glue	Menerapkan batas 500 karakter untuk nilai-nilai variabel substitusi.
Vertica	Amazon Redshift	Konversi yang diimplementasikan dari BINARY VARBINARY LONG BINARY, BYTEA,,, dan tipe RAW data ke tipe VARBYTE data.
Vertica	Amazon Redshift	Peningkatan konversi fungsi dan literal bawaan.

Catatan rilis untuk AWS SCT Build 661

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan filter untuk mencari aturan pemetaan dalam tampilan pemetaan. Saat Anda menerapkan filter, AWS SCT menampilkan aturan yang cocok dengan kondisi pemfilteran dalam daftar pemetaan Server. Untuk informasi selengkapnya, lihat Mengelola aturan pemetaan .
Semua	Semua	Upgrade Apache Log4j ke versi 2.17.1.
Semua	Amazon Redshift	Menambahkan dukungan migrasi data ke Amazon Redshift menggunakan ENCRYPTED klausa dalam perintah. COPY
Semua	Amazon Redshift	Meningkatkan REST API dari agen ekstraksi data. REST API yang diperbarui menambahkan dukungan properti baru seperti kunci enkripsi, jenis enkripsi, dan sebagainya.
Semua	Amazon Redshift	Menerapkan peran dengan asumsi dalam agen ekstraksi data. Pembaruan ini meningkatkan distribusi subtugas, dan memungkinkan AWS SCT untuk menetapkan tugas ke agen bebas dari peran yang ditentukan.
Semua	Amazon Redshift	Menerapkan pemeriksaan bahwa semua komponen yang diperlukan diinstal sebelum paket ekstensi diterapkan ke Amazon Redshift.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi fungsi ERROR_LINE ,ERROR_MESSAGE , ERROR_NUMBER ERROR_PROCEDURE ,ERROR_SEVERITY , dan ERROR_STATE sistem untuk penanganan kesalahan.
Microsoft SQL Server DW		
IBM Db2 untuk z/OS	Aurora MySQL	Menambahkan dukungan IBM Db2 untuk z/OS versi 12 sebagai sumber untuk proyek migrasi database di. AWS SCT Untuk informasi selengkapnya, lihat Menggunakan IBM Db2 untuk z/OS sebagai sumber .

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
	Aurora PostgreSQL L MySQL PostgreSQL L	
IBM Db2 LUW	Semua	Meningkatkan pemuat metadata sumber untuk memastikan bahwa AWS SCT memuat parameter rutin yang menduplikasi nama kolom.
Database Microsoft Azure SQL	Aurora PostgreSQL L	Memperbaiki kesalahan transformator untuk prosedur dengan pernyataan yang SET NOCOUNT ON ditetapkan.
Microsoft SQL Server	PostgreSQL L	
Database Microsoft Azure SQL	Aurora PostgreSQL L	Peningkatan konversi CONCAT fungsi ketika nilai input adalah variabel dari tipe yang ditentukan pengguna.
Microsoft SQL Server	PostgreSQL L	
Database Microsoft Azure SQL	Aurora PostgreSQL L	Menyelesaikan masalah di mana DATEPART fungsi tersebut salah dikonversi.
Microsoft SQL Server	PostgreSQL L	

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Menerapkan dukungan versi baru dari file konfigurasi fitur Babelfish . File ini mendefinisikan fitur SQL yang didukung dan tidak didukung oleh versi Babelfish tertentu.
Microsoft SQL Server DW	Amazon Redshift	Menyelesaikan masalah di mana prosedur dengan EXECUTE pernyataan salah dikonversi.
Microsoft SSIS	AWS Glue	Meningkatkan antarmuka pengguna wizard konfigurasi pekerjaan. AWS SCT sekarang hanya menampilkan koneksi yang tersedia di bagian konfigurasi koneksi.
Microsoft SSIS	AWS Glue	Menyelesaikan masalah di mana aturan transformasi tidak diterapkan pada tugas paket dan aturan variabel.
Microsoft SSIS	AWS Glue AWS Glue Studio	Menambahkan item tindakan baru 25042 untuk komponen yang tidak didukung.
Microsoft SSIS	AWS Glue Studio	Konversi yang diimplementasikan dari Microsoft SQL Server Integration Services (SSIS) mengekstrak, mengubah, dan memuat paket (ETL) ke paket. AWS Glue Studio Untuk informasi selengkapnya, lihat Mengonversi SSIS keAWS Glue Studio .
Oracle	MariaDB	Memperbaiki masalah dengan konversi MINUS operator.
Oracle	MariaDB	Peningkatan konversi dariROWNUM,, SYS_GUIDTO_CHAR, dan ADD_MONTHS fungsi ketika variabel sql_mode sistem di MariaDB adalah untuk Oracle.
Oracle	PostgreSQL	Menambahkan opsi untuk menghindari konversi jenis variabel bind ke tipe SQL dalam proyek konversi aplikasi generik.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL	Menambahkan opsi untuk menghindari menambahkan nama skema ke nama objek yang dikonversi dalam proyek konversi aplikasi generik.
Oracle	PostgreSQL	Ditambahkan dukungan dari format variabel ?x bind untuk aplikasi konversi kode SQL.
Oracle DW	Amazon Redshift	Konversi yang diimplementasikan dari tipe RAW data ke tipe VARBYTE data.
Teradata	Amazon Redshift	Ditambahkan pilihan untuk meniru SET tabel dalam kode dikonversi. Untuk emulasi ini, AWS SCT dukungan MIN dan MAX kondisi.
Teradata	Amazon Redshift	Peningkatan konversi operasi gabungan yang memiliki parameter tipe data yang berbeda. Pembaruan ini memungkinkan AWS SCT untuk menerapkan aturan transformasi selama konversi operasi tersebut.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana GROUP BY klausa salah dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana QUALIFY klausa salah dikonversi.
Teradata	Amazon Redshift	Teratasi kesalahan tak terduga terjadi selama impor FastExport skrip.
Teradata	Amazon Redshift RSQL	Menerapkan kemampuan untuk mengedit nilai-nilai variabel dalam Teradata BTEQ dan skrip shell.
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah di mana skrip manifes tidak ada untuk sesi Teradata FastLoad yang dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah di mana ekstensi file manifes hilang di pencari sumber daya seragam (URL) untuk FastLoad skrip yang dikonversi.
Teradata BTEQ	Amazon Redshift RSQL	Memperbaiki kesalahan loader untuk skrip dengan variabel substitusi.
Teradata BTEQ	Amazon Redshift RSQL	Memperbaiki masalah saat item tindakan 27022 tidak muncul saat diperlukan.

Catatan rilis untuk AWS SCT Build 660

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan dukungan AWS Secrets Manager dan Secure Sockets Layer (SSL) dalam laporan penilaian multiserver. Untuk informasi selengkapnya, lihat Membuat laporan penilaian multiserver untuk migrasi database .
Semua	Semua	Peningkatan koleksi statistik untuk objek dikonversi.
Semua	PostgreSQL	Menerapkan dukungan PostgreSQL mayor versi 14 dan MariaDB 10.6 sebagai target migrasi.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan logika transformasi untuk nama-nama objek yang dikonversi.
Database Microsoft Azure SQL	Aurora PostgreSQL	Peningkatan konversi tipe XML data.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server		
Database Microsoft Azure SQL	Aurora PostgreSQL	Menyelesaikan masalah di mana NOT LIKE klausa salah dikonversi.
Microsoft SQL Server	PostgreSQL	
Database Microsoft Azure SQL	Aurora PostgreSQL	Memperbaiki kesalahan transformator untuk prosedur dengan INSERT, DELETE, dan UPDATE pernyataan yang menyertakan OUTPUT klausa.
Microsoft SQL Server	PostgreSQL	
Database Microsoft Azure SQL	Aurora PostgreSQL	Memperbaiki kesalahan transformator untuk prosedur dengan RETURN @@ROWCOUNT pernyataan.
Microsoft SQL Server	PostgreSQL	
Microsoft SQL Server	Semua	Peningkatan konversi prosedur yang menggunakan server tertaut.
Microsoft SQL Server	Semua	Menambahkan dukungan Microsoft Windows Authentication dalam laporan penilaian multiserver.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Memperbaiki kesalahan transformator untuk konstruktor nilai tabel.
Microsoft SQL Server DW	Amazon Redshift dan AWS Glue	Peningkatan konversi skrip ekstrak, transformasi, dan muat (ETL) untuk menyertakan jalur yang benar ke skrip yang dikonversi.
Microsoft SQL Server DW	Amazon Redshift	Menyelesaikan masalah di mana skrip yang dikonversi yang berbeda dibuat untuk platform basis data target virtual dan nyata.
Oracle	PostgreSQL Aurora PostgreSQL	Menambahkan dukungan untuk konversi indeks untuk tampilan terwujud.
Oracle	PostgreSQL Aurora PostgreSQL	Memperbaiki masalah saat item tindakan 5982 tidak muncul saat mengonversi PRIMARY KEY dan UNIQUE membatasi opsi. NOVALIDATE
Oracle DW	Amazon Redshift	Menyelesaikan masalah di mana kategori tambahan ditampilkan dalam skema yang dikonversi.
Teradata	Amazon Redshift	Memperbaiki masalah saat item tindakan 13185 tidak muncul saat mengonversi kolom yang belum terselesaikan sebagai argumen fungsi. CAST

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Peningkatan konversi DELETE dan DELETE ALL pernyataan untuk menggunakan TRUNCATE perintah dalam kode yang dikonversi.
Teradata	Amazon Redshift	Peningkatan konversi SET tabel.
Teradata	Amazon Redshift	Peningkatan konversi NORMALIZE kondisi.
Teradata	Amazon Redshift	Memperbarui laporan penilaian untuk menghapus statistik konversi skema database dari daftar objek penyimpanan database.
Teradata	Amazon Redshift	Peningkatan konversi UPDATE pernyataan tanpa FROM klausa.
Teradata	Amazon Redshift	Dukungan yang diimplementasikan dari tipe VARBYTE data dalam kode yang dikonversi.
Teradata BTEQ	AWS Glue	Menyelesaikan masalah saat AWS Glue opsi Convert to dinonaktifkan di menu konteks.
Teradata BTEQ	Amazon Redshift RSQL	Menyelesaikan masalah di mana tipe data hilang dalam kode yang dikonversi.
Teradata BTEQ	Amazon Redshift RSQL	Menyelesaikan masalah di mana variabel substitusi salah dikutip dalam kode yang dikonversi.
Teradata BTEQ	Amazon Redshift RSQL	Memperbaiki masalah dengan konversi variabel substitusi dengan nilai dalam FastLoad skrip.
Vertica	Amazon Redshift	Dukungan yang diimplementasikan dari tipe TIME data dalam kode yang dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Vertica	Amazon Redshift	Peningkatan konversi SELECT DISTINCT dan ORDER BY ekspresi.
Vertica	Amazon Redshift	Ditambahkan dukungan untuk konversi kendala.
Vertica	Amazon Redshift	Mengatasi kesalahan saat laporan penilaian tidak disimpan sebagai file nilai dipisahkan koma (CSV).

Catatan rilis untuk AWS SCT Build 659

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Memperbaiki wizard proyek Baru yang menghasilkan laporan penilaian gabungan untuk beberapa basis data sumber.
Semua	Semua	Memperbaiki masalah saat paket ekstensi tidak dibuat dalam proyek yang menyertakan beberapa basis data sumber dan target.
Semua	Semua	Peningkatan konversi kode SQL yang tertanam dalam kode sumber aplikasi.
Semua	Semua	Menambahkan kemampuan untuk menjalankan skrip dari folder yang berbeda di antarmuka baris AWS SCT perintah.
Semua	Amazon Redshift	Memperbaiki pesan peringatan yang diberikan saat pengguna memilih Jalankan pengoptimalan dalam proyek migrasi dengan platform basis data target virtual Amazon Redshift.
Semua	Aurora PostgreSQL	Menerapkan dukungan PostgreSQL mayor versi 13 pada Aurora PostgreSQL Compatible Edition sebagai target migrasi.
Semua	Amazon RDS for MySQL	Menerapkan konversi kode case insensitive secara default.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Analisis Sinaps Azure	Amazon Redshift	Mengatasi kesalahan di mana koneksi ke database sumber gagal di antarmuka baris perintah.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Peningkatan konversi prosedur yang mencakup UPDATE pernyataan dengan kondisi gabungan.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Peningkatan konversi pemacu, prosedur tersimpan, dan fungsi yang mencakup nilai setelah tanda sama dengan.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Memperbaiki kesalahan transformator untuk prosedur dengan DELETE pernyataan dan OR operator.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Peningkatan konversi OUTPUT klausa.
Microsoft SQL Server DW	Amazon Redshift dan AWS Glue	Peningkatan konversi tipe NUMERIC data.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server DW	Amazon Redshift	Peningkatan konversi tampilan yang memiliki alias tabel dengan nama yang sama dengan tabel asli.
Microsoft SSIS	AWS Glue	Memperbaiki masalah saat kredensi AWS Glue koneksi tidak ditampilkan di jendela Konfigurasi koneksi.
Netezza	Amazon Redshift	Menambahkan kemampuan untuk mengulangi menjalankan tugas migrasi data change data capture (CDC) setiap hari.
Netezza	Amazon Redshift	Memperbaiki masalah saat tab Tugas menjadi tidak aktif setelah membatalkan pendaftaran agen ekstraksi data.
Netezza	Amazon Redshift	Memperbaiki masalah saat konfirmasi pendaftaran agen migrasi data tidak ditampilkan di antarmuka pengguna.
Netezza	Amazon Redshift	Memperbaiki masalah saat koneksi ke database sumber gagal dengan kesalahan Loader.
Netezza	Amazon Redshift	Mengatasi kesalahan di mana agen migrasi data gagal dijalankan setelah membuka proyek yang disimpan.
Oracle	Amazon RDS for Oracle	Menerapkan dukungan Oracle Unified Audit.
Oracle	PostgreSQL Aurora PostgreSQL	Menerapkan konversi kode SQL dalam aplikasi C#. Untuk informasi selengkapnya, lihat Mengonversi kode SQL dalam aplikasi C# .

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL L Aurora PostgreSQL L	Menerapkan logika transformasi baru untuk nama objek peka huruf besar/kecil untuk meningkatkan visibilitas perubahan konversi kode. AWS SCT mengubah nama objek dalam huruf besar menjadi huruf kecil. Kebalikannya juga benar; AWS SCT mengonversi nama objek dalam huruf kecil menjadi huruf besar. Nama objek lain dan kata-kata yang dipesan dikonversi tanpa perubahan.
Oracle	PostgreSQL L Aurora PostgreSQL L	Peningkatan konversi partisi hash tanpa kendala. NOT NULL
Oracle	Aurora PostgreSQL L	Ditambahkan dukungan untuk konversi Oracle CHECK, FOREIGN KEY, dan NOT NULL kendala dengan klausa. ENABLE NOVALIDATE
Oracle DW	Amazon Redshift	Memperbaiki masalah di mana nilai yang salah untuk nomor floating point dimigrasikan.
Oracle DW	Amazon Redshift dan AWS Glue	Menyelesaikan masalah dengan kolom kosong dalam laporan penilaian migrasi database dalam file nilai dipisahkan koma (CSV).
SAP ASE	PostgreSQL L Aurora PostgreSQL L	Memperbaiki masalah dengan gangguan konversi yang tidak terduga.
Kepingan salju	Amazon Redshift	Peningkatan konversi tipe VARIANT data.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Peningkatan konversi COLLECT STATISTICS pernyataan.
Teradata	Amazon Redshift	Memperbaiki masalah saat item tindakan 9998 tidak muncul saat mengonversi tampilan bersarang dengan kolom. PERIOD
Teradata	Amazon Redshift dan AWS Glue	Memperbaiki masalah saat platform AWS Glue target virtual tidak ditampilkan di UI setelah membuka proyek yang disimpan.
Teradata BTEQ	AWS Glue	Memperbaiki masalah saat konversi ke platform AWS Glue target virtual tidak didukung setelah membuka proyek yang disimpan.
Teradata BTEQ	Amazon Redshift RSQL	Penyorotan sintaks yang ditingkatkan dari kode yang dikonversi.
Teradata BTEQ	Amazon Redshift RSQL	Diimplementasikan memeriksa nilai parameter setelah upload. Nilai yang tidak didukung disorot pada tab Variabel.
Vertica	Amazon Redshift	Konversi fungsi agregat yang diimplementasikan.
Vertica	Amazon Redshift	Menerapkan konversi proyeksi ke tampilan terwujud dan meningkatkan UI yang menampilkan kode sumber proyeksi.

Catatan rilis untuk AWS SCT Build 658

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menyediakan integrasi dengan AWS Secrets Manager. Anda sekarang dapat menggunakan kredensial koneksi database yang disimpan di Secrets Manager.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan dukungan untuk skrip dalam format YAMAL di antarmuka baris AWS SCT perintah.
Semua	Amazon Redshift	Menerapkan dukungan titik akhir antarmuka Amazon S3 (VPCE) di agen ekstraksi data.
Semua	Amazon Redshift	Menambahkan dukungan untuk platform basis data target virtual Amazon Redshift selain Amazon AWS Glue Redshift dan kombinasi yang sudah didukung.
Greenplum	Amazon Redshift	Memperbaiki masalah di mana opsi Simpan sebagai SQL tidak menyimpan kode SQL yang dikonversi ke file.
IBM Db2 LUW Database Microsoft Azure SQL Microsoft SQL Server Oracle SAP ASE	Aurora MySQL	Peningkatan konversi untuk mendukung fitur baru Amazon Aurora MySQL Edisi yang kompatibel dengan MySQL dengan kompatibilitas MySQL 8.0.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Memperbaiki masalah saat item tindakan 810 tidak muncul saat diperlukan.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Peningkatan konversi prosedur dengan UPDATE, DELETE, dan INSERT pernyataan.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Memperbaiki masalah saat item tindakan 7810 tidak muncul saat diperlukan.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Peningkatan konversi EXEC pernyataan yang bersarang di dalam IF...ELSE pernyataan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi tampilan yang diindeks.
Netezza	Amazon Redshift	Agensi migrasi data yang ditingkatkan dengan melacak transaksi langsung selama pemuatan penuh dalam operasi pengambilan data perubahan (CDC). Anda sekarang dapat menghentikan tugas migrasi data jika sesi CDC dijadwalkan untuk dimulai pada waktu tertentu. Selain itu, Anda dapat melihat tingkat pencatatan kesalahan di konsol setelah Anda menghentikan tugas dengan CDC.
Oracle	Semua	Meningkatkan pemuat tabel untuk memastikan bahwa AWS SCT memuat objek dengan opsi berbagi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi SYSDATE fungsi dan menambahkan kemampuan untuk mengubah zona waktu dalam pengaturan Konversi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Menyelesaikan masalah di mana pernyataan dinamis tidak dikonversi.
Oracle	Aurora PostgreSQL L PostgreSQL L	Memperbaiki masalah di mana kode yang dikonversi tidak menyertakan nama yang dihasilkan sistem.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle Oracle DW	Aurora PostgreSQL PostgreSQL	Peningkatan konversi SELECT pernyataan yang bersarang di dalam pemicu.
Oracle DW	Amazon Redshift	Peningkatan konversi TO_DATE, TO_TIMESTAMP , dan TO_TIMESTAMP_TZ fungsi dalam paket ekstensi.
Kepingan salju	Amazon Redshift	Ditambahkan pilihan untuk menyimpan kode SQL dikonversi dalam file yang berbeda untuk setiap objek atau untuk setiap pernyataan.
Teradata	Amazon Redshift	Peningkatan konversi CONCAT fungsi.
Teradata	Amazon Redshift	Peningkatan konversi SELECT pernyataan yang bersarang di dalam WHERE klausa.
Teradata	Amazon Redshift	Menyelesaikan masalah saat SET dan MULTISSET tabel salah dikonversi setelah pengguna menjatuhkan dan membuat ulang tabel.
Teradata	Amazon Redshift	Peningkatan konversi prosedur yang mencakup WITH klausa.
Teradata	Amazon Redshift	Peningkatan konversi tipe DATE data.
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah saat terjadi kesalahan transformator tak terduga selama konversi FastExport skrip.
Teradata BTEQ	Amazon Redshift RSQL	Menambahkan dukungan untuk konversi indeks gabungan ke tampilan terwujud.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata BTEQ	Amazon Redshift RSQL	Menambahkan dukungan untuk konversi TITLE definisi yang mencakup beberapa baris.
Teradata BTEQ	Amazon Redshift RSQL	Menyelesaikan masalah di mana ukuran tipe data geospasial tidak dikonversi.
Teradata BTEQ	Amazon Redshift RSQL	Memperbaiki masalah di mana nama parameter dikonversi ke karakter huruf kecil.
Teradata BTEQ	Amazon Redshift RSQL	Memperbaiki masalah saat prosedur tersimpan yang bersarang di dalam MACRO pernyataan tidak dikonversi.
Vertica	Amazon Redshift	Peningkatan konversi ALL operator.
Vertica	Amazon Redshift	Menyelesaikan masalah di mana Use Union all view? opsi dalam pengaturan Konversi tidak diterapkan.
Vertica	Amazon Redshift	Peningkatan konversi dari TIME dan tipe TIME WITH TIMEZONE data.
Vertica	Amazon Redshift	Menyelesaikan masalah dengan pemuatan tabel fleksibel.

Masalah terselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 657

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Upgrade Apache Log4j ke versi 2.17 untuk mengatasi masalah kerentanan keamanan.
Semua	Amazon Redshift	Proyek pengoptimalan skema yang ditingkatkan, di mana statistik manajemen kunci tidak disimpan dalam proyek. AWS SCT
Amazon Redshift	Amazon Redshift	Memperbaiki masalah dengan pembaruan informasi server.
Apache Cassandra	Amazon DynamoDB	Memperbaiki masalah dengan aturan pemetaan saat menggunakan antarmuka baris AWS SCT perintah.
Apache Cassandra	Amazon DynamoDB	Menyelesaikan masalah saat tugas migrasi tidak dibuat karena judul yang diperbarui dalam sertifikat.
Microsoft SQL Server	Aurora PostgreSQL	Memperbaiki masalah sehingga item tindakan 7672 tidak muncul selama konversi prosedur Microsoft SQL Server dengan SQL dinamis.
Database Azure SQL	Aurora PostgreSQL	Peningkatan konversi fungsi bernilai tabel.
Microsoft SQL Server	PostgreSQL	
Database Azure SQL	Aurora PostgreSQL	Menyelesaikan masalah di mana OUT argumen dalam prosedur tersimpan dengan nilai pengembalian default tidak dikonversi ke INOUT argumen.
Microsoft SQL Server	PostgreSQL	

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Greenplum	Amazon Redshift	Meningkatkan strategi optimasi dengan menemukan tabel dan kolom yang paling banyak digunakan dari QueryLog tabel.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Memperbaiki masalah dengan konversi berikut ini: <ul style="list-style-type: none"> • Operator penugasan penggabungan string () += • Fungsi SCOPE_IDENTITY • Tipe data varchar(max)
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Peningkatan konversi tampilan dengan fungsi yang tidak didukung.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Memperbaiki masalah di mana fungsi yang tidak didukung sebagai argumen ke fungsi lain salah dikonversi.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Peningkatan konversi referensi tabel transisi.
Microsoft SQL Server DW	Amazon Redshift	Menambahkan kategori fungsi agregat ke pohon metadata database sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server DW	Amazon Redshift	Peningkatan konversi tipe TIME data.
Analisis Sinaps Azure Greenplum Netezza Microsoft SQL Server DW Kepingan salju Teradata	Amazon Redshift	Memperbaiki masalah di mana DROP dan CREATE skrip tidak disimpan saat menggunakan platform basis data target virtual.
Layanan Integrasi Microsoft SQL Server	AWS Glue	Menyelesaikan masalah di mana skrip objek sumber tidak ditampilkan di UI.
Netezza	Amazon Redshift	Meningkatkan strategi optimasi dengan memilih tabel fakta dan dimensi yang sesuai untuk kolokasi.
Oracle	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah untuk mengonversi pemicu Oracle dengan benar, yang menggunakan nomor urut.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL L PostgreSQL L	Peningkatan konversi tampilan dengan link database publik.
Oracle DW	Amazon Redshift	Meningkatkan strategi optimasi dengan menganalisis kardinalitas kolom indeks.
Oracle DW	Amazon Redshift	Memperbaiki masalah saat fungsi skalar yang ditentukan pengguna khusus dengan rangkaian string tidak dikonversi dengan benar.
Kepingan salju	Amazon Redshift	Memperbaiki masalah di mana opsi Simpan sebagai SQL tidak ditampilkan di UI.
Teradata	Amazon Redshift	Memperbaiki masalah saat pengumpulan statistik gagal dengan LOADER ERROR pengecualian.
Teradata	Amazon Redshift	Memperbaiki masalah saat opsi Buat laporan tidak ditampilkan di UI.
Teradata	Amazon Redshift	Peningkatan konversi CAST fungsi.
Teradata	Amazon Redshift	Memperbaiki konversi yang rusak untuk <code>ST_Line_Interpolate_Point</code> .
Teradata	Amazon Redshift	Menghapus nilai tak terduga dari jalur pustaka Python.
Teradata	Amazon Redshift RSQL	Memperbaiki kesalahan resolver yang muncul selama konversi beberapa FastLoad skrip.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata BTEQ	Amazon Redshift RSQL	Peningkatan konversi tipe data DATABASE perintah dan geometri.
Teradata BTEQ	AWS Glue	Memperbaiki masalah dengan sinkronisasi skrip sumber dan target yang salah di UI.

Masalah terselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 656

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan dukungan dari beberapa sumber dan basis data target dalam satu proyek. Pengguna sekarang dapat membuat aturan pemetaan untuk mencocokkan skema database yang berbeda dan platform target dalam proyek yang sama.
Semua	Semua	Menambahkan dukungan platform basis data target virtual. Pengguna sekarang tidak perlu terhubung ke database target untuk melihat bagaimana AWS SCT mengubah skema basis data sumber mereka.
Semua	Semua	Perbaikan UI: <ul style="list-style-type: none"> • Menambahkan Connect ke server dan Disconnect dari opsi server ke pohon metadata sumber dan target. • Menambahkan opsi untuk menghapus server database dari AWS SCT proyek.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Cassandra	Amazon DynamoDB	Menyelesaikan masalah pencarian di mana CASSANDRA_HOME variabel tidak menyertakan garis miring (/) setelah cassandra .yaml atau conf folder.
Cassandra	Amazon DynamoDB	Menambahkan dukungan Amazon Machine Image (AMI) untuk Amazon Linux 2.
Cassandra	Amazon DynamoDB	Pesan kesalahan yang ditingkatkan diberikan ketika kunci yang salah diberikan untuk Cassandra.
Cassandra	Amazon DynamoDB	Peningkatan konversi dengan mengubah properti dalam cassandra -env .yaml file tergantung pada versi database target.
Cassandra	Amazon DynamoDB	Meningkatkan versi Java pada target Cassandra Datacenter ke 1.8.0.
Greenplum	Amazon Redshift	Strategi pengoptimalan yang ditingkatkan di Pengaturan Proyek.
Greenplum	Amazon Redshift	Menyelesaikan masalah migrasi data di mana objek tidak diterapkan ke database dengan kesalahan ini: An I/O error occurred while sending to the backend .
Greenplum Microsoft SQL Server DW	Amazon Redshift	Menyelesaikan masalah di mana Apply RTRIM to string columns opsi tidak ditampilkan di UI.
Microsoft SQL Server	Babelfish for Aurora PostgreSQL	Menambahkan dukungan untuk Babelfish untuk Aurora PostgreSQL sebagai platform target. Pengguna sekarang dapat membuat laporan penilaian untuk memperkirakan migrasi dari SQL Server ke Babelfish untuk Aurora PostgreSQL.
Netezza	Amazon Redshift	Strategi pengoptimalan yang ditingkatkan di Pengaturan Proyek.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SAP ASE	Aurora PostgreSQL PostgreSQL	Menerapkan kemampuan untuk menghasilkan nama unik untuk indeks.
SAP ASE	Aurora PostgreSQL PostgreSQL	Memperbaiki masalah dengan kolom indeks duplikat dalam skrip target.
Kepingan salju	Amazon Redshift	Menyelesaikan masalah di mana Sembunyikan skema kosong, Sembunyikan database kosong, dan opsi Sembunyikan database sistem/skema tidak ditampilkan di UI.
Teradata	Amazon Redshift RSQL	Menambahkan dukungan untuk konversi skrip MultiLoad pekerjaan Teradata ke skrip Amazon Redshift RSQL.
Teradata	Amazon Redshift RSQL	Memperbaiki masalah dengan konversi variabel substitusi dalam FastLoad dan FastExport skrip.
Teradata	Amazon Redshift RSQL	Memperbaiki masalah saat item tindakan tidak ditampilkan di tab Item Tindakan setelah beralih dari tab Ringkasan.
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah saat terjadi kesalahan setelah menghasilkan laporan selama konversi FastExport skrip.
Teradata	Amazon Redshift RSQL	Mengatasi masalah pemformatan setelah konversi skrip shell.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift RSQL	Memperbaiki masalah sehingga AI 13177 sekarang dikomentari dalam skrip yang dikonversi.
Teradata	Amazon Redshift	Memperbaiki konversi tabel temporal yang rusak.
Teradata	Amazon Redshift	Peningkatan konversi SET QUERY_BAND pernyataan.
Teradata	Amazon Redshift	Memperbaiki konversi NORMALIZE operasi yang rusak.
Vertica	Amazon Redshift	Memperbaiki deskripsi AI 17008.

Masalah terselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 655

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift RSQL	Memperbaiki masalah untuk memastikan semua masalah penilaian muncul dalam laporan saat FastLoad atau MultiLoad digunakan.
Teradata	Amazon Redshift RSQL	Menambahkan dukungan untuk konversi skrip FastExport pekerjaan Teradata ke skrip Amazon Redshift RSQL.
Teradata	Amazon Redshift RSQL	Memperbaiki masalah untuk memastikan tindakan Simpan manifes ke S3 diaktifkan dalam mode offline saat menggunakan FastLoad.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift RSQL	Memperbaiki masalah untuk memastikan aturan pemetaan diterapkan untuk skrip seperti. FastLoad
Greenplum	Amazon Redshift	Meningkatkan versi driver minimum yang didukung untuk Greenplum menjadi 42.2.5.
Greenplum	Amazon Redshift	Menambahkan koneksi ke Greenplum melalui SSL dengan driver versi 42.2.5 atau lebih tinggi.
Oracle DW	Amazon Redshift	Peningkatan dukungan untuk mengeksekusi fungsi skalar yang ditentukan pengguna kustom (UDF) dalam UDF lain.
Oracle DW	Amazon Redshift	Memperbaiki masalah saat fungsi tidak diterapkan ke database dengan kesalahan ini: <code>Failed to compile udf</code> .
Oracle DW	Amazon Redshift	Peningkatan konversi dengan menggunakan deklarasi tipe yang sesuai seperti, <code>pls-type</code> untuk parameter <code>% ROWTYPE</code> .
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah saat masalah penilaian tipe informasi tidak ditampilkan dalam laporan.
Teradata	Amazon Redshift RSQL	Mengatasi kesalahan transformator setelah mengonversi beberapa skrip.
Teradata	Amazon Redshift RSQL	Memperbaiki masalah sehingga masalah sekarang dikomentari dalam skrip yang dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana <code>FastExport ->EXPORT -> 'null'</code> ditampilkan sebagai gantinya <code>'CAST'</code> setelah konversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana beberapa fungsi paket ekstensi gagal saat diterapkan <code>Cause: [JDBC Driver]String index out of range: 0</code> jika menggunakan driver versi 1.2.43

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Konversi tabel SET — Tetapkan emulasi tabel ditambahkan untuk pernyataan insert-select.
Teradata	Amazon Redshift	Cast—mendukung pengecoran tipe data tambahan.
Teradata	Amazon Redshift	Memperbaiki konversi yang rusak untuk "other_current_time_01"
Teradata	Amazon Redshift	Teradata FastExport - Amazon Redshift RSQL: Peningkatan konversi perintah Teradata — Bidang FastExport
Teradata	Amazon Redshift	Teradata FastExport - Amazon Redshift RSQL: Peningkatan konversi perintah Teradata — Layout FastExport
Oracle	PostgreSQL Aurora PostgreSQL	Menyelesaikan masalah di mana skrip target objek dengan PERNYATAAN PENGECUALIAN SIMPAN berubah setelah konversi ulang.
Oracle	PostgreSQL Aurora PostgreSQL	Menyelesaikan masalah di mana bidang yang salah ditentukan dalam klausa ORDER BY setelah proc_cursor_with_calc_columns konversi.
Oracle	PostgreSQL Aurora PostgreSQL	Terselesaikan: deklarasi aws_oracle_ext\$array_id\$temporary variabel tambahan diperlukan dalam konversi KOLEKSI ASOSIATIF.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL L	Terselesaikan: konversi yang salah dari PRIMARY KEY dengan nama yang sama dari INDEX yang dimiliki oleh tabel yang sama.
	Aurora PostgreSQL L	

Masalah terselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 654

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL L	Menyelesaikan masalah dengan Hierarchical Query Pseudocolumns, kesalahan penguraian kolom PRIOR.
	Aurora PostgreSQL L	
Oracle	PostgreSQL L	Menyelesaikan masalah untuk mengonversi komentar multi-baris dengan benar yang berisi garis miring dan tanda bintang (/ *).
	Aurora PostgreSQL L	
Oracle	PostgreSQL L	Menambahkan tampilan sistem USER_COL_COMMENTS emulasi ke paket ekstensi.
	Aurora PostgreSQL L	

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL Aurora PostgreSQL	Peningkatan konversi literal yang dikutip.
DB2 LUW	PostgreSQL Aurora PostgreSQL	Peningkatan konversi pernyataan LABEL yang menambah atau mengganti label dalam deskripsi tabel, tampilan, alias, atau kolom.
Oracle	Tidak ada	Tabel sistem SYS.USER\$ diganti dengan tampilan DBA_USERS, dan kueri yang ditingkatkan.
Oracle DW	Amazon Redshift	Kueri metadata Oracle DW yang diperbarui.
Teradata	Amazon Redshift RSQL	Menambahkan dukungan untuk konversi skrip shell, Teradata FastLoad, dan Teradata Basic Teradata Query (BTEQ) ke skrip Amazon Redshift RSQL.
Teradata BTEQ	Amazon Redshift RSQL	Masalah teratasi di mana "merge_01" salah dikonversi.
Teradata BTEQ	Amazon Redshift RSQL	Masalah teratasi sehingga End or Identify (EOI) muncul di akhir skrip pada baris baru.
Sinaps Azure	Amazon Redshift	Pesan kesalahan yang ditingkatkan diberikan ketika kata sandi yang salah diberikan untuk Azure Synapse.
Teradata	Amazon Redshift	Peningkatan konversi pernyataan UPDATE untuk meneruskan nama alias yang tepat per standar Teradata.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Mengatasi kesalahan konversi kursor di mana tindakan tidak diterima.
Teradata	Amazon Redshift	Menyelesaikan masalah saat konversi TD_NORMALIZE_OVERLAP menjatuhkan baris.
Teradata	Amazon Redshift	Sekarang mendukung pemeriksaan tanggal yang ketat untuk fungsi TO_DATE yang disempurnakan.
Teradata	Amazon Redshift	Peningkatan konversi fungsi Built-in TO_NUMBER (n).
Teradata	Amazon Redshift	Menyelesaikan masalah di mana kategori Skema tidak ada di pohon metadata.
Greenplum	Amazon Redshift	Menambahkan pilihan GP_SEGMENT_ID ke daftar saat membuat partisi virtual untuk tabel Greenplum.
Greenplum	Amazon Redshift	Menyelesaikan masalah di mana fungsi tidak diterapkan pada target.
MS SQL Server DW	Amazon Redshift	Menyelesaikan masalah di mana kesalahan transformasi terjadi setelah konversi tanpa AI 9996.
MS SQL Server DW	Amazon Redshift	Menyelesaikan masalah saat kesalahan dicatat saat membuka wizard paket ekstensi.
MS SQL Server DW	Amazon Redshift	Menyelesaikan masalah ketika gaya komentar yang salah digunakan untuk fungsi Redshift Python.
Netezza	Amazon Redshift	Menyelesaikan masalah di mana paket ekstensi Netezza—Redshift dengan profil gagal dibuat. AWS
Teradata	Amazon Redshift RSQL	Peningkatan konversi perintah FastLoad SESSIONS.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift RSQL	Laporan penilaian FastLoad skrip yang ditingkatkan.
Teradata	Amazon Redshift RSQL	Menerapkan FastLoad WRITER Simpan ke tindakan S3.
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah di mana tombol FastLoad Simpan Skrip\ Simpan manifes ke s3 tidak aktif.
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah di mana FastLoad multifile_script hanya membuat satu file manifes setelah konversi, bukan tiga file yang diharapkan.
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah di mana FastLoad folder tambahan ditampilkan di jalur S3.
Teradata	Amazon Redshift RSQL	Menyelesaikan masalah yang FastLoad memiliki nama file manifes yang salah di jalur S3.

Masalah terselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 653

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL	Menerapkan kemampuan untuk mengonversi SQL dinamis yang dibuat dalam fungsi atau prosedur yang disebut.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
	Aurora PostgreSQL	
Oracle	PostgreSQL Aurora PostgreSQL	Peningkatan konversi SQL Dinamis: Dalam parameter sebagai variabel mengikat.
Oracle DW 18, 19	Amazon Redshift	Peningkatan konversi Oracle ke Redshift diterapkan: peningkatan konversi bawaan. Agregat LISTAGG; LISTAGG analitik.
Oracle DW 18,19	Amazon Redshift	Peningkatan konversi Oracle ke Redshift diterapkan: Kueri fitur baru.
Vertica	Amazon Redshift	Peningkatan konversi Vertica ke Redshift diterapkan: koneksi SSL ke JDBC dengan SSL = True.
MS SQL Server DW	Amazon Redshift	MS SQL Server untuk perbaikan konversi Redshift: Tabel Eksternal.
Teradata	Amazon Redshift	Peningkatan konversi Teradata ke Redshift: operasi aritmatika tipe data INTERVAL.
Teradata	Amazon Redshift	Peningkatan konversi Teradata ke Redshift: Dukungan untuk alias kolom lateral.
Oracle	Tidak ada	Kueri Loader berikut sekarang digunakan DBA_USERS sebagai pengganti: SYS.USER\$ <ul style="list-style-type: none"> • get-tree-path-list- by-name-path .sql • estimate-table-or-view- constraints-by-schema .sql • estimate-table-or-view- constraints-by-selected-schemas .sql

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Peningkatan penyelarasan komentar saat SCT mengonversi makro Teradata ke prosedur tersimpan Redshift.
Oracle DW	Amazon Redshift	Peningkatan konversi elemen format Tanggal/Timestamp:., dan TO_DATE TO_TIMESTAMP TO_TIMESTAMP_TZ
Teradata	Amazon Redshift	Teratasi kesalahan konversi kursor Teradata.
Teradata	Amazon Redshift	Masalah teratasi yang menyebabkan atribut TD_NORMALIZE_OVERLAP dijatuhkan selama konversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana MAX fungsi diabaikan saat SCT mengonversi kueri.
Teradata	Amazon Redshift	SCT sekarang mengkonversi fungsi Karakter Teradata ke fungsi Redshift LENGTH.
Teradata	Amazon Redshift	SCT sekarang mendukung konversi FORMAT ke TO_CHAR untuk format yang paling umum digunakan.
Semua	Semua	Peningkatan konversi rutinitas terenkripsi.

Masalah terselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 652

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	PostgreSQL	Menambahkan penguncian aplikasi sp_getapplock dan sp_releaseapplock fungsi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Tidak ada	Amazon Redshift	Peningkatan Command Line Interface (CLI): menerapkan mode Script Command.
Oracle	PostgreSQL Aurora PostgreSQL	Diimplementasikan parameter rutin pengambilan sampel di dalam SQL dinamis.
Oracle	PostgreSQL Aurora PostgreSQL	Perbaikan konversi ke SQL dinamis yang dibuat dalam fungsi atau prosedur yang disebut.
Microsoft SQL Server Oracle DB2 LUW	Aurora PostgreSQL	Setiap fungsi lambda diterapkan dan dikonfigurasi melalui kebijakan hanya sekali, dan fungsi lambda umum digunakan kembali untuk semua sumber yang memungkinkan.
DB2 LUW	PostgreSQL	Masalah teratasi yang menyebabkan pesan kesalahan, “9996 - Keperahan kritis - Kesalahan transformator terjadi” saat menggunakan DB2 LUW sebagai sumber.
Teradata	Amazon Redshift	Support untuk ekspresi tabel rekursif dalam peluncuran Amazon Redshift yang akan datang.
Sinaps Azure	Amazon Redshift	Aturan optimasi skema yang diterapkan.
Teradata	Amazon Redshift	Mendukung konversi Zona Waktu dari makro Teradata ke prosedur tersimpan Redshift.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Support aritmatika pada nilai PERIODE.
Teradata	Amazon Redshift	Support konversi ekspresi tabel umum rekursif Teradata (RECURSIVE CTE).
Teradata	Amazon Redshift	Support case sensitive identifier melalui pengaturan pengguna, <code>enable_case_sensitive_identifier</code> . Jadi, "COLUMN_NAME" dan "Column_Name" menjadi nama kolom yang berbeda.
Teradata	Amazon Redshift	Mengatasi masalah tipe data desimal sehingga bidang Desimal dikonversi dengan presisi yang sama.
Teradata	Amazon Redshift	Mengatasi masalah dengan konversi aritmatika interval sehingga pengurangan aritmatika interval dikonversi dengan benar.
Teradata	Amazon Redshift	Peningkatan Nomor Teradata ke pengecoran tipe DATE.
Teradata	Amazon Redshift	Peningkatan Tanggal Teradata ke pengecoran tipe NUMBER
Teradata BTEQ	Amazon Redshift	Peningkatan konversi tipe data PERIODE.
Teradata	Amazon Redshift	Masalah yang diselesaikan dengan memuat metadata untuk tabel dengan kolom GEOMETRI sehingga sekarang dimuat dari Teradata dengan benar.
Teradata	Amazon Redshift	Support konversi pernyataan gabungan saat mengonversi makro Teradata ke prosedur tersimpan Redshift.
Teradata	Amazon Redshift	Peningkatan konversi makro sederhana saat bermigrasi dari Teradata ke Redshift.
Teradata	Amazon Redshift	Memastikan konversi pernyataan Teradata UPDATE meneruskan nama alias yang tepat sesuai standar Teradata.

Masalah terselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 651

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	AWS SCT Laporan yang disempurnakan untuk memperbarui tautan ke item tindakan konversi yang direkomendasikan yang terdaftar.
MS SQL Server	PostgreSQL	Ditambahkan dukungan untuk konversi STR() fungsi.
MS SQL Server	PostgreSQL	Menambahkan dukungan untuk mengonversi operator EXOR bitwise (di ^ Microsoft SQL Server) ke PostgreSQL sebagai operator. #
Oracle	PostgreSQL	Menyelesaikan masalah di mana <code>aws_oracle_ext.UNI STR(null)</code> fungsi paket AWS SCT ekstensi tergantung NULL pada target PostgreSQL. AWS SCT sekarang menangani NULL.
Teradata BTEQ	Amazon Redshift RSQL	Peningkatan konversi dilakukan untuk menyelesaikan masalah saat konversi Amazon Redshift RSQL MERGE memberikan kesalahan transformasi.
Oracle DW	Amazon Redshift	Menerapkan built-in yang ditingkatkan.
Oracle DW	Amazon Redshift	Menambahkan penyempurnaan berbasis fitur metadata, termasuk partisi Daftar Otomatis (TBL_PART_LIST_AUTO), Daftar Multi-kolom (TBL_PART_MULTI_LIST) dan Interval-Reference (TBL_PART_RANGE_INTVAL_REF).
none	Amazon Redshift	Peningkatan batas tabel partisi partisi fisik yang digunakan untuk UNION ALL konversi.
Teradata	Amazon Redshift	Perbaikan konversi dilakukan pada ruang lingkup laporan Penilaian.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Perbaikan konversi dilakukan untuk konversi MACRO Teradata yang kompleks.
Teradata	Amazon Redshift	Peningkatan konversi makro Teradata ke prosedur tersimpan Amazon Redshift sambil mengomentari SQL yang tidak didukung.
Teradata	Amazon Redshift	Menyelesaikan masalah saat konversi makro Teradata ke prosedur tersimpan Amazon Redshift menghasilkan referensi nama alias yang salah.
Teradata	Amazon Redshift	Peningkatan konversi pernyataan TeradataQUALIFY.
Teradata	Amazon Redshift	Peningkatan konversi untuk meneruskan komentar ke Amazon Redshift dan mempertahankan riwayat perubahan yang dilakukan pada tampilan.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana klausa RESET WHEN tidak menghasilkan konversi yang benar.
Teradata BTEQ	Amazon Redshift	Peningkatan konversi skrip BTEQ yang berisi pernyataan MERGE.
Teradata	Amazon Redshift	Menambahkan fungsi bawaan untuk meningkatkan konversi bidang tipe data PERIODE.
Microsoft SQL Server	Amazon Redshift	Pemetaan tipe data transformasi yang disempurnakan untuk tipe data TIME.
Semua	Semua	Menambahkan akses ke publikasi awal manual Referensi AWS Schema Conversion Tool CLI dalam format PDF. Lihat AWS Schema Conversion Tool Referensi CLI .

Masalah terselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 650

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	<p>Penggunaan agen ekstraktor yang diperbarui dan ditingkatkan, termasuk:</p> <ul style="list-style-type: none"> Konfigurasi untuk digunakan dengan penyimpanan bersama dan agen penyalinan khusus. Mengekspor dan mengimpor tugas ekstraksi data dari satu proyek ke proyek lainnya. Support untuk Azure SQL Data Warehouse (Azure Synapse) sebagai sumber. Menggunakan partisi Netezza asli. <p>Untuk informasi selengkapnya, lihat Memigrasi data dari gudang data lokal ke Amazon Redshift.</p>
Semua	Amazon RDS PostgreSQL 13	AWS SCT sekarang mendukung Amazon RDS PostgreSQL 13 sebagai target.
Microsoft SQL Server	Aurora PostgreSQL	Peningkatan konversi hasil yang ditetapkan dari prosedur Microsoft SQL Server ke target PostgreSQL Aurora.
Oracle DW	Amazon Redshift	Menerapkan peningkatan konversi Oracle ke Amazon Redshift.
Oracle DW	Amazon Redshift	Peningkatan yang diterapkan untuk mengonversi pernyataan SQL dinamis.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle DW	Amazon Redshift	Menerapkan perbaikan untuk konversi SQL UDF.
Oracle DW	Amazon Redshift	Pesan yang diklarifikasi yang AWS SCT tidak mendukung konversi TABEL EKSTERNAL.
Oracle DW	Amazon Redshift	Fungsi konversi bawaan yang disempurnakan.
Teradata BTEQ	Amazon Redshift RSQL	Peningkatan parameter substitusi penanganan di dalam skrip BTEQ saat menggunakan GUI. AWS SCT
Microsoft SQL Server DW	Semua	Upgrade versi driver JDBC minimum yang didukung untuk Microsoft SQL Server, Azure, Azure Synapse.
Microsoft SQL Server		
Biru langit		
Sinaps Azure		

Masalah terselesaikan:

- Teradata: Peningkatan tambahan konversi makro [DISELESAIKAN]
- Karakter khusus lolos dalam target menyebabkan kesalahan SQL dan pekerjaan ulang diperlukan untuk menempatkannya kembali [RESOLVED]
- Perbaikan umum

Catatan rilis untuk AWS SCT Build 649

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server DW	Amazon Redshift	Peningkatan konversi MSSQL ke Amazon Redshift untuk mendukung tabel temporal.
Oracle DW	Amazon Redshift	<p>Peningkatan fungsi bawaan yang diterapkan, seperti:</p> <p>Fungsi konversi</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
Oracle DW	Amazon Redshift	<p>Peningkatan fungsi yang diterapkan untuk Perkiraan Pemrosesan Kueri, seperti:</p> <p>Fungsi agregat</p> <ul style="list-style-type: none"> • NILAI APA PUN • APPROX_COUNT_DISTINCT

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		<ul style="list-style-type: none"> • KIRA-KIRA_COUNT_DISTINCT_DETAIL • KIRA-KIRA_COUNT_DISTINCT_AGG • LISTAGG • TO_APPROX_COUNT_DISTINCT
Teradata	Amazon Redshift	Peningkatan konversi yang diterapkan untuk pemilihan kunci pengurutan dan distribusi otomatis Teradata. Mesin DB secara otomatis memilih distribusi dan mengurutkan kunci. Memperkenalkan tombol radio berlabel Gunakan penyetelan tabel otomatis Amazon Redshift ke pengaturan proyek saat ini > Strategi pengoptimalan > dialog Strategi Pemilihan Kunci Awal.
Teradata	Amazon Redshift	Pemuat AWS SCT tabel yang disempurnakan untuk memastikan AWS SCT memuat semua tabel dari Teradata.
Teradata	Amazon Redshift	Peningkatan konversi yang diterapkan sehingga Amazon Redshift mendukung pola subquery berkorelasi yang menyertakan klausa WHERE NOT EXISTS sederhana.
Teradata	Amazon Redshift	Menambahkan dukungan untuk penggunaan perintah ECHO di makro.
DB2 LUW	PostgreSQL	Dukungan yang diterapkan untuk konversi SET HASIL DINAMIS, termasuk:
	Aurora PostgreSQL	<ul style="list-style-type: none"> • Klausa kursor DENGAN PENGEMBALIAN/DENGAN KEMBALI KE KLIEN • HASIL DINAMIS MENETAPKAN konversi klausa rutin

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server Oracle DB2 LUW SAP ASE	Aurora PostgreSQL	Menerapkan dukungan untuk Aurora RDS PostgreSQL saat ini sebagai target.
Microsoft SQL Server Oracle DB2 LUW SAP ASE	MariaDB	Menerapkan dukungan untuk MariaDB 10.5 sebagai target.
Microsoft SQL Server	MariaDB	Dukungan yang diimplementasikan dari INSERT-RETURNING yang mengembalikan set hasil dari baris yang disisipkan.
Oracle	Aurora PostgreSQL	Ditambahkan dukungan fungsi XMLFOREST untuk mengkonversi dari Oracle ke Aurora PostgreSQL.

Masalah terselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 648

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL	Aurora PostgreSQL extension pack custom apply mode diimplementasikan: operator untuk tipe numerik/tanggal dan teks.
	Edisi yang Kompatibel dengan Amazon Aurora PostgreSQL	
Oracle Microsoft SQL Server DB2 LUW	Aurora PostgreSQL	<p>Konfigurasi Aurora PostgreSQL Lambda Invoke diimplementasikan: pembuatan ekstensi <code>aws_lambda</code>; penetapan peran IAM ke cluster PostgreSQL Aurora.</p> <ul style="list-style-type: none"> • Oracle — Email, Pekerjaan,, Antrian, File WebAgent • DB2—Email, Tugas, File • Microsoft SQL Server— Email, Agen
Oracle	PostgreSQL	<p>Refactoring konversi pernyataan FORALL diimplementasikan:</p> <ul style="list-style-type: none"> • Pernyataan FORALL • UNTUK SEMUA... SIMPAN PENGECEUALIAN • KEMBALI KE DENGAN KOLEKSI MASSAL • Koleksi sistem SQL%BULK_EXCEPTIONS

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle DW 18, 19	Amazon Redshift	Peningkatan konversi Oracle ke Amazon Redshift diterapkan: peningkatan konversi bawaan. Agregat LISTAGG; LISTAGG analitik.
Oracle DW 18,19	Amazon Redshift	Peningkatan konversi Oracle ke Amazon Redshift diterapkan: Kueri fitur baru.
Vertica	Amazon Redshift	Peningkatan konversi Vertica ke Amazon Redshift diterapkan: koneksi SSL ke JDBC dengan SSL = True.
Microsoft SQL Server DW	Amazon Redshift	Peningkatan konversi Microsoft SQL Server ke Redshift: Tabel Eksternal.
Teradata	Amazon Redshift	Peningkatan konversi Teradata ke Redshift: operasi aritmatika tipe data INTERVAL.
Teradata	Amazon Redshift	Peningkatan konversi Teradata ke Redshift: Dukungan untuk alias kolom lateral.

Masalah terselesaikan:

- Perbaikan umum

Catatan rilis untuk AWS SCT Build 647

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	Microsoft SQL Server	RDS sekarang mendukung fitur Database Mail.
Microsoft SQL Server	MySQL	Menerapkan nama maksimum setiap jenis pengenal - Panjang maksimum nama objek (misalnya, tabel, kendala, kolom) di SQL Server adalah 128 karakter. Panjang maksimum nama objek di MySQL adalah 64 karakter. Untuk menulis objek dikonversi ke

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		<p>database MySQL Anda perlu mempersingkat nama mereka. Untuk mencegah nama duplikat setelah memotong, Anda perlu menambahkan “checksum” dari nama objek asli ke nama baru.</p> <p>Potong nama lebih dari 64 karakter sebagai berikut:</p> <pre>[first N chars]() + "" + [checksum]()</pre> <pre>[first N chars] = 64 - 1 - [length of checksum string]</pre> <p>Sebagai contoh:</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	Diimplementasikan beban dan konversi komentar pada objek penyimpanan. Misalnya, pemrosesan komentar pada Tabel, dan pemrosesan komentar pada kolom Tabel/Tampilan.
Teradata	Amazon Redshift	Menambahkan dukungan untuk konversi tipe data TIME.
Teradata	Amazon Redshift	Peningkatan konversi - TD_NORMALIZE_OVERLAP diimplementasikan.
Microsoft SQL Server DW	Amazon Redshift	Peningkatan konversi - PILIH dengan klausa WITH; PILIH tanpa DARI
Semua	Semua	AWS SCT Data Migration Service Assessor (DMSA) - Fitur baru ini memungkinkan Anda mengevaluasi beberapa server dan menerima laporan ringkasan yang menunjukkan arah target terbaik untuk lingkungan Anda.
Semua	Semua	AWS SCT Wizard - Perbandingan target sekarang menunjukkan perbedaan antara target dalam tampilan tabel tunggal.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	UI Filter Pohon - Filter metadata yang didesain ulang menangani pola penyaringan yang lebih kompleks.
Semua	Semua	Laporan Penilaian - Bagian Peringatan yang Dirancang Ulang memberikan deskripsi yang lebih baik dan pemahaman yang lebih jelas tentang suatu masalah.

Masalah terselesaikan:

- Perbaikan umum
- Ekstraktor Data — Subtugas gagal dengan ConcurrentModificationException [RESOLVED].
- Microsoft SQL Server ke MySQL - panjang pengenalan maks [DISELESAIKAN].

Catatan rilis untuk AWS SCT Build 646

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL L	Implementasi model format TM yang ditingkatkan.
Oracle	PostgreSQL L	Implementasi masker format SP memberikan dukungan dasar untuk akhiran SP, hanya untuk bahasa Inggris.
Oracle	PostgreSQL L	Penanganan nama objek panjang Oracle - AWS SCT sekarang menangani nama objek panjang Oracle sesuai dengan atribut panjang pengenalan maksimum target.
	Amazon Redshift	Amazon Redshift encoding AZ64 dengan AWS SCT - Menambahkan pengkodean kompresi AZ64 untuk beberapa tipe data
Teradata	Amazon Redshift	Menambahkan dukungan untuk konversi transaksi Implisit.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Menambahkan dukungan untuk fungsi built-in geospasial Teradata: Metode <code>ST_LineString</code>
Greenplum	Amazon Redshift	Konversi urutan Greenplum - Menambahkan item berikutnya ke tab Properti: nilai min, nilai maks, kenaikan, siklus.
Greenplum	Amazon Redshift	Resolver - Menambahkan pemecahan tipe data "char".
Greenplum	Amazon Redshift	Panjang konversi karakter - Diperbarui konversi PL/PGSQL untuk jenis karakter.
Greenplum	Amazon Redshift	Menyelesaikan masalah dengan pemilihan kunci distribusi Greenplum di mana tabel memiliki KUNCI DISTRIBUSI tetapi AWS SCT tidak dapat mengenali dan mengambil tabel sebagai DIDISTRIBUSIKAN SECARA ACAK.
Teradata	Amazon Redshift	Dukungan kursor Teradata - Ditambahkan dukungan untuk konversi kursor.
Teradata	Amazon Redshift	Kolom Identitas - Ditambahkan dukungan untuk konversi kolom Identitas.
Teradata	Amazon Redshift	Jenis data INTERVAL - Menambahkan dukungan untuk konversi tipe data INTERVAL.

Masalah terselesaikan:

- Perbaikan umum
- Greenplum: Tidak dapat menjalankan konversi karena kesalahan dalam log [DISELESAIKAN].
- MSSQL - PostgreSQL: Kesalahan transformator saat mengonversi fungsi LAG [DISELESAIKAN].
- MSSQL - PostgreSQL: SCOPE_IDENTITY [DISELESAIKAN].
- AWS SCT tergantung di proyek DW [RESOLVED].
- Perlu aturan pemetaan untuk menghapus ruang tambahan pada nama kolom di AWS SCT [RESOLVED].

Catatan rilis untuk AWS SCT Build 645

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Berikan solusi untuk menyelesaikan tampilan Teradata yang tidak memenuhi syarat sepenuhnya (melihat nama atau objek yang tidak memenuhi syarat sepenuhnya dalam tampilan).
Teradata	Amazon Redshift	Ditambahkan dukungan fungsi ASCII untuk menghitung node.
Teradata	Amazon Redshift	Saat AWS SCT melihat data multi-byte dalam Teradata yang CHAR didefinisikan sebagai CHAR(N), data tersebut dikonversi menjadi di Amazon VARCHAR(3*N) Redshift.
Teradata	Amazon Redshift	Berikan CAST konversi Teradata antara tanggal dan angka. <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 konversi tipe PERIOD data Teradata menjadi dua kolom Amazon TIMESTAMP Redshift: <ul style="list-style-type: none"> <code>PERIOD(TIMESTAMP)</code> <code>PERIOD(TIMESTAMP WITH TIMEZONE)</code>
Teradata	Amazon Redshift	Support konversi RANK fungsi Teradata dengan RESET WHEN klausa.
Teradata	Amazon Redshift	Peningkatan dukungan CAST dalam konversi tipe data eksplisit, dan CAST implisit pada ekspresi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Laporkan pola subquery berkorelasi yang tidak didukung. Untuk informasi selengkapnya, lihat Subkueri berkorelasi di Panduan Pengembang Database Amazon Redshift.
tidak ada	Amazon Redshift	Tabel yang ditingkatkan membatasi dukungan untuk tipe node RA3.
Teradata	Amazon Redshift	Menambahkan dukungan untuk ekstraksi data geospasial Teradata. Untuk informasi selengkapnya, lihat Mengkueri data spasial di Amazon Redshift di Panduan Pengembang Database Amazon Redshift.
Microsoft SQL Server	PostgreSQL	Ditambahkan pilihan, <code>convert_procedures_to_function</code> .

Masalah terselesaikan:

- Perbaikan umum

Catatan rilis untuk AWS SCT Build 644

Perubahan untuk AWS SCT rilis 1.0.643 digabungkan menjadi rilis 1.0.644. AWS SCT

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Beberapa peningkatan konversi. <ul style="list-style-type: none"> • Peningkatan konversi dengan QUALIFY dengan alias tabel. • Konversi yang ditingkatkan dengan IN operator. • Peningkatan konversi dengan LIKE operator. •

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		<p>Konversi yang ditingkatkan dengan menyoroti masalah dalam kode yang dikonversi.</p> <ul style="list-style-type: none">• Peningkatan konversi dengan urutan yang tidak biasa dari WHERE, QUALIFY klausa dalam SQL.• Kesalahan transformator tetap terjadi selama konversi JOIN() konstruksi prosedur UPD_FT_SVC_TRANS_BH_CBH_IND .• Peningkatan konversi makro ke prosedur yang disimpan. <p>Menambahkan perintah AWS SCT CLI khusus yang dapat mengurai skrip sql/bteq yang disediakan dan menghasilkan laporan tentang jumlah struktur sintaks yang ditemui dalam kode sumber.</p> <ul style="list-style-type: none">• Hitungan perintah BTEQ• Hitungan HANDLERS• Hitungan kasus CAST• Hitungan kasus DML/DDDL• Hitungan DML pada tampilan yang dapat diperbarui <p>Menambahkan item tindakan laporan penilaian: Kolom Teradata dengan format tanggal khusus tidak didukung di Amazon Redshift.</p>

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Menambahkan fungsionalitas untuk menyimpan skrip instalasi paket ekstensi.</p> <p>Tingkat keparahan berubah untuk AI 5334.</p> <p>Peningkatan kinerja menggunakan catatan sebagai variabel paketIMPLEMENTATION .</p> <p>Ditambahkan XMLAGG dukungan fungsi agregasi</p>
IBM Db2	PostgreSQL/Aurora PostgreSQL	Menambahkan beban dan konversi komentar pada implementasi objek penyimpanan.
MS SQL DW	Amazon Redshift	<p>Peningkatan konversi: Masalah yang diselesaikan denganPATINDEX.</p> <p>Perbaiki UI:</p> <ul style="list-style-type: none"> • Simpan sebagai SQL untuk implementasi pohon sumber. • Menambahkan logika tambahan untuk pembuatan skrip untuk beberapa file.
Vertica	Amazon Redshift	Peningkatan UI: Simpan sebagai SQL untuk implementasi pohon sumber.

Masalah terselesaikan:

- Perbaikan umum untuk konversi antara Teradata dan Amazon Redshift
- Perbaikan bug umum dan peningkatan UI

Catatan rilis untuk AWS SCT Build 642

Perubahan untuk AWS Schema Conversion Tool rilis 1.0.642.

Note

AWS Schema Conversion Tool (AWS SCT) perubahan build 1.0.642 berlaku untuk Windows, Ubuntu, dan Fedora. Tidak ada build 1.0.642 untuk macOS.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SSIS	AWS Glue	Menerapkan konversi paket ETL Microsoft SQL Server Integration Services (SSIS) ke. AWS Glue Untuk informasi selengkapnya, lihat Mengonversi SSIS keAWS GluebersamaAWS SCT .
Oracle	Mode MariaDB/S QL=Oracle /MySQL/ Amazon Aurora MySQL	Menerapkan bagian deklarasi PL/SQL dalam klausa WITH.
Oracle	PostgreSQL/Aurora PostgreSQL	Menambahkan dukungan untuk <code>DBMS_SESSION.RESET_PACKAGE</code> dan <code>DBMS_SESSION.MODIFY_PACKAGE</code> .
Vertica	Amazon Redshift	Aktifkan ekspor skrip SQL dari database Vertica ke Amazon Redshift.

Masalah terselesaikan:

- Peningkatan Laporan Penilaian.
- Peningkatan UI Laporan Penilaian.
- Tambahkan kemampuan untuk mengubah pengaturan JVM dari UI.
- Perbaikan umum.

Catatan rilis untuk AWS SCT build 641

Perubahan untuk AWS Schema Conversion Tool rilis 1.0.641.

Note

AWS Schema Conversion Tool (AWS SCT) perubahan build 1.0.641 berlaku untuk Windows, Ubuntu, dan Fedora. Tidak ada build 1.0.641 untuk macOS.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle/ MS SQL/ MySQL/ PostgreSQL/ DB2 LUW	Semua	Menghasilkan perhitungan Laporan Waktu dalam file.csv.
Teradata	Amazon Redshift	Ditambahkan dukungan untuk fungsi CSUM. Menambahkan dukungan untuk tipe data geospasial Teradata.
Teradata	Semua	Ditambahkan dukungan untuk mengkonversi kolom IDENTITY.
Greenplum	Amazon Redshift	Menambahkan dukungan untuk gaya distribusi AUTO selama konversi tabel Greenplum.
SAP ASE	Semua	Menghasilkan perhitungan Laporan Waktu dalam file.csv.

Terselesaikan:

- Berbagai perbaikan bug
- Berbagai peningkatan kinerja.

Catatan rilis untuk AWS SCT Build 640

Perubahan untuk AWS SCT rilis 1.0.633, 1.0.634, 1.0.635, 1.0.636, 1.0.637, 1.0.638, 1.0.639, dan 1.0.640 digabungkan menjadi rilis 1.0.640. AWS SCT

Note

AWS SCT perubahan build 1.0.640 berlaku untuk Windows, Ubuntu, dan Fedora. Mereka tidak berlaku untuk macOS.

Anda tidak dapat menginstal AWS SCT versi 1.0.640 atau lebih tinggi di Apple macOS. AWS SCT versi 1.0.632 adalah versi terakhir yang mendukung instalasi di Apple macOS.

Dalam tabel berikut, Anda dapat menemukan daftar fitur dan perbaikan bug untuk AWS Schema Conversion Tool versi yang telah digabungkan menjadi rilis 1.0.640. Tabel ini mengelompokkan fitur dan perbaikan bug oleh mesin sumber.

Topik

- [Rilis 1.0.640 perubahan Oracle](#)
- [Rilis 1.0.640 Microsoft SQL Server perubahan](#)
- [Rilis 1.0.640 Perubahan MySQL](#)
- [Rilis 1.0.640 perubahan PostgreSQL](#)
- [Rilis 1.0.640 Db2 perubahan LUW](#)
- [Rilis 1.0.640 Perubahan Teradata](#)
- [Rilis 1.0.640 perubahan untuk mesin lain](#)

Rilis 1.0.640 perubahan Oracle

Daftar tabel berikut membangun 1.0.640 perubahan di mana Oracle adalah mesin sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL	Menerapkan konversi kode SQL dalam aplikasi Java dan Pro* C.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
	Aurora PostgreSQL	
Oracle	PostgreSQL Aurora PostgreSQL	<p>Peningkatan kinerja fungsi berikut saat digunakan dalam klausa WHERE:</p> <ul style="list-style-type: none"> • <code>aws_oracle_ext.to_date</code> • <code>aws_oracle_ext.to_char</code> • <code>aws_oracle_ext.to_number</code> • <code>aws_oracle_ext.sysdate</code> • <code>aws_oracle_ext.sys_context</code>
Oracle	RDS MariaDB 10.4	Menambahkan dukungan RDS MariaDB 10.4 untuk semua vendor pemrosesan transaksional online (OLTP).
Oracle	PostgreSQL L/Aurora PostgreSQL L	<p>Ditambahkan dukungan untuk <code>DBMS_UTILITY.GET_TIME</code>.</p> <p>Menambahkan emulasi berikut:</p> <ul style="list-style-type: none"> • <code>DBMS_UTILITY.GET_TIME</code> • <code>DBMS_UTILITY.FORMAT_CALL_STACK</code> • <code>DBMS_UTILITY.CURRENT_INSTANCE</code>

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	MariaDB/MySQL/Aurora MySQL/Microsoft SQL Mode Server = Oracle/PostgreSQL/Aurora PostgreSQL/RDS Oracle	Menambahkan dukungan klausa berbagi untuk TABLE (DATA, EXTENDED DATA), VIEW (DATA, EXTENDED DATA), dan SEQUENCE (DATA)
Oracle	PostgreSQL/Aurora PostgreSQL/Oracle RDS	<p>Definisi DEFAULT kolom dapat diperluas agar DEFAULT diterapkan untuk penyisipan NULL eksplisit.</p> <p>Klausa DEFAULT memiliki klausa ON NULL baru. Klausa baru ini menginstruksikan database untuk menetapkan nilai kolom default tertentu ketika pernyataan INSERT mencoba untuk menetapkan nilai yang mengevaluasi ke NULL.</p>
Oracle	MariaDB/MariaDB (MODE SQL = ORACLE)	Ditambahkan dukungan untuk “Kolom Identitas,” yang secara otomatis meningkat pada saat penyisipan.
Semua	Semua	Tingkatkan ke Amazon Corretto JDK 11 dari JDK 8. Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	MariaDB 10.2/MariaDB 10.3/ MySQL/ Aurora MySQL/ PostgreSQL/ Aurora PostgreSQL	DEFAULTKlausula memiliki ON NULL klausula baru, yang menginstruksikan database untuk menetapkan nilai kolom default tertentu ketika pernyataan INSERT mencoba untuk menetapkan nilai yang dievaluasi. NULL
Oracle	Oracle RDS/ MySQL /Aurora MySQL/ PostgreSQL/ Aurora PostgreSQL	Ditambahkan dukungan untuk IDENTITY kolom.
Oracle	MySQL 8.x	Ditambahkan dukungan untuk kendala CHECK.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Diimplementasikan memeriksa ANYDATA IS NULL/BUKAN NULL menggunakan paket ekstensi rutin.</p> <p>Menerapkan emulasi fungsi VALUE yang digunakan dalam kueri berdasarkan fungsi TABLE dari XMLSequence.</p> <p>Menambahkan dukungan DBMS_LOB untuk rutinitas bawaan berikut:</p> <ul style="list-style-type: none"> • DBMS_LOB.CREATETEMPORARY • DBMS_LOB.FREETEMPORARY • DBMS_LOB.APPEND
Semua	SQL Server	<p>SQL Server 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: Menambahkan dukungan untuk Graph Databases Node dan tipe tabel Edge.</p> <p>SQL Server 2016: Menambahkan dukungan untuk TABEL TEMPORAL.</p>
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.
Oracle	Amazon Redshift	<p>Menerapkan konversi atribut kursor di blok bersarang.</p> <p>Amazon Redshift tidak mendukung koleksi. Variabel terkait dikonversi sebagai VARCHAR. Semua operasi pengumpulan selain menetapkan satu variabel ke variabel lain ditolak, termasuk inisiasi dan akses elemen koleksi.</p> <p>Menerapkan gaya distribusi Amazon Redshift = AUTO.</p>

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Jika kata nonreserved di Oracle dicadangkan di PostgreSQL, maka berikut ini benar:</p> <ul style="list-style-type: none"> • Jika kata tersebut dikutip, ia mempertahankan kasusnya dan tetap dikutip. • Jika kata itu tidak dikutip, itu dilemparkan ke huruf besar dan dikutip. <p>Menerapkan kemampuan untuk menggunakan fungsi sebagai input ke fungsi LTRIM, RTRIM, dan TRIM.</p> <p>PILIH DISTINCT, ORDER BY ekspresi harus muncul di daftar pilih.</p> <p>Untuk parameter kursor yang mengikuti setelah parameter dengan nilai DEFAULT, AWS SCT tambahkan klausa DEFAULT IS NULL</p> <p>Parameter kursor OUT sumber dikonversi ke parameter kursor IN.</p> <p>Variabel paket diimplementasikan kembali dengan menambahkan opsi “Package variable logic implementation” di bawah “Conversion settings”. Pengaturan yang tersedia adalah: “variabel sesi” dan “objek global plv8”. Defaultnya adalah “variabel sesi”.</p> <p>Menerapkan dukungan pragma AUTONOMOUS_TRANSACTION dengan dblink dan pg_background.</p>
Oracle	Semua	Diimplementasikan tampilan SYS_%_TAB_COMMENTS.
Oracle	PostgreSQL	Masukan variabel ke filter tidak didukung di PostgreSQL. Saat mengonversi dari Oracle ke PostgreSQL, jika filter variabel ditemui pengecualian sekarang dilaporkan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Amazon Redshift	<p>Menerapkan kode tersimpan FOR.. LOOP Peningkatan konversi kursor.</p> <p>Diimplementasikan pemanggilan kode tersimpan fungsi/prosedur dengan parameter default.</p> <p>Menerapkan kemampuan kode tersimpan untuk UPDATE dengan alias tanpa klausa WHERE.</p> <p>Fungsi kode tersimpan yang diimplementasikan membentuk kasus tambahan dengan SELECT FROM dual.</p> <p>Diimplementasikan tabel kode tersimpan %parameter RowType dan variabel paket.</p> <p>Diimplementasikan kode tersimpan yang digunakan dari JAVA dan prosedur eksternal.</p> <p>Diimplementasikan paket Oracle standar dalam kode yang disimpan.</p>

Rilis 1.0.640 Microsoft SQL Server perubahan

Daftar tabel berikut membangun 1.0.640 perubahan di mana Microsoft SQL Server adalah mesin sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft Azure/Microsoft SQL Server	PostgreSQL/Aurora PostgreSQL/MySQL/Aurora MySQL	Ditambahkan dukungan untuk indeks COLUMN STORE.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server	RDS MariaDB 10.4	Menambahkan dukungan RDS MariaDB 10.4 untuk semua vendor pemrosesan transaksional online (OLTP).
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Menambahkan dukungan untuk atribut indeks OPTIMIZE_FOR_SEQUENTIAL_KEY.
Azure/SQL Server	MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Menambahkan dukungan untuk Database Node dan tipe tabel Edge.
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Ditambahkan dukungan untuk TEMPORAL TABLES.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Tingkatkan ke Amazon Corretto JDK 11 dari JDK 8. Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
Azure/SQL Server	MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL/MariaDB	Ditambahkan dukungan untuk pengolahan DHTML untuk SQL Server Graph Architecture.
SQL Server	Aurora PostgreSQL	Ditambahkan pilihan untuk mengkonversi parameter tanpa par_ awalan.
Azure/SQL Server	MySQL 8.x	Ditambahkan dukungan untuk kendala CHECK.
Semua	SQL Server	<p>SQL Server 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: Menambahkan dukungan untuk Graph Databases Node dan tipe tabel Edge.</p> <p>SQL Server 2016: Menambahkan dukungan untuk TABEL TEMPORAL.</p>
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQL Server	AWS Glue (Cangkang Python)	<p>Perbaiki konversi, termasuk:</p> <ul style="list-style-type: none"> Menerapkan konversi fungsi bawaan ke Python.String. Diimplementasikan EXECUTE dan EXEC dalam kode yang disimpan. Diimplementasikan menggunakan jenis tabel.
Azure/SQL Server	PostgreSQL/Aurora PostgreSQL	Diimplementasikan membuat prosedur \$TMP opsional.
SQL Server	MySQL/Aurora MySQL	<p>Operasi aritmatika yang diperpanjang dengan tanggal.</p> <p>Emulasi konstruksi 'TOP (ekspresi) DENGAN IKATAN.</p> <p>Setelah memanggil prosedur dengan refkursor yang dihasilkan keluar, refkursor sekarang ditutup.</p> <p>Menyetel tingkat isolasi GLOBAL tidak didukung di Aurora MySQL. Hanya lingkup sesi yang dapat diubah. Perilaku default transaksi adalah menggunakan REPEATABLE READ dan pembacaan yang konsisten. Aplikasi yang dirancang untuk digunakan dengan READ COMMITTED mungkin perlu dimodifikasi. Atau, mereka dapat secara eksplisit mengubah default menjadi READ COMMITTED.</p>

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQL Server	AWS Glue (Canggih Python)	<p>Pernyataan SQL Server menghasilkan set hasil yang lengkap, tetapi ada kalanya hasilnya paling baik diproses satu baris pada satu waktu. Membuka kursor pada set hasil memungkinkan pemrosesan hasil yang ditetapkan satu baris pada satu waktu. Anda dapat menetapkan kursor ke variabel atau parameter dengan tipe data kursor.</p> <p>Diimplementasikan melampirkan serangkaian pernyataan Transact-SQL untuk kode yang disimpan sehingga sekelompok pernyataan Transact-SQL dapat dijalankan meskipun Python tidak mendukung SQL Server BEGIN dan END sebagai control-of-flow</p> <p>SQL Server LABEL dan pernyataan GOTO tidak didukung oleh AWS Glue. Jika AWS SCT menemukan label dalam kode, itu dilewati. Jika AWS SCT menemukan pernyataan GOTO, itu dikomentari.</p>
SQL Server	Amazon Redshift	<p>Menerapkan pemrosesan kondisional pernyataan Transact-SQL untuk kode yang disimpan dengan mengimplementasikan IF... Kontrol ELSE.</p> <p>Diimplementasikan melampirkan serangkaian pernyataan Transact-SQL untuk kode yang disimpan sehingga sekelompok pernyataan Transact-SQL dapat dijalankan sebagai blok. Mendukung BEGIN bersarang... Blok END.</p> <p>Diimplementasikan SET dan SELECT dalam kode yang disimpan.</p> <p>Menerapkan CREATE INDEX di Amazon Redshift (yang tidak mendukung indeks) dengan membuat kunci pengurutan yang ditentukan pengguna pada tabel.</p>

Rilis 1.0.640 Perubahan MySQL

Daftar tabel berikut membangun 1.0.640 perubahan di mana MySQL adalah mesin sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
MySQL	PostgreSQL 12.x	Ditambahkan dukungan untuk kolom yang dihasilkan.
Semua	Semua	Tingkatkan ke Amazon Corretto JDK 11 dari JDK 8. Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
MySQL	PostgreSQL/Aurora PostgreSQL 11.	Ditambahkan dukungan untuk yang berikut: <ul style="list-style-type: none"> • Transaksi tertanam di dalam prosedur tersimpan SQL. • Kemampuan untuk MEMANGGIL prosedur tersimpan SQL. • Kemampuan untuk membuat prosedur tersimpan SQL.
Semua	SQL Server	<p>SQL Server 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: Menambahkan dukungan untuk Graph Databases Node dan tipe tabel Edge.</p> <p>SQL Server 2016: Menambahkan dukungan untuk TABEL TEMPORAL.</p>
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.

Rilis 1.0.640 perubahan PostgreSQL

Tabel berikut mencantumkan perubahan build 1.0.640 di mana PostgreSQL adalah mesin sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
PostgreSQL	MySQL 8.x	<p>MySQL sekarang mendukung pembuatan bagian kunci indeks fungsional yang mengindeks nilai ekspresi daripada nilai kolom. Bagian kunci fungsional memungkinkan pengindeksan nilai, seperti nilai JSON, yang tidak dapat diindeks sebaliknya.</p> <p>MySQL sekarang mendukung Now CTE dan Recursive CTE.</p>
Semua	Semua	Tingkatkan ke Amazon Corretto JDK 11 dari JDK 8. Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
PostgreSQL 11.x	PostgreSQL/Aurora PostgreSQL 11.	<p>Ditambahkan dukungan untuk yang berikut:</p> <ul style="list-style-type: none"> • Transaksi tertanam di dalam prosedur tersimpan SQL. • Kemampuan untuk MEMANGGIL prosedur tersimpan SQL. • Kemampuan untuk membuat prosedur tersimpan SQL.
PostgreSQL	MySQL 8.x	<p>Menambahkan dukungan MySQL untuk indeks menurun. DESC dalam definisi indeks tidak lagi diabaikan, tetapi menyebabkan penyimpanan nilai-nilai kunci dalam urutan menurun.</p> <p>Menambahkan MySQL mendukung penggunaan ekspresi sebagai nilai default dalam spesifikasi tipe data, termasuk ekspresi sebagai nilai default untuk tipe data BLOB, TEXT, GEOMETRY, dan JSON.</p> <p>Beberapa fungsi agregat yang ada sekarang dapat digunakan sebagai fungsi jendela:</p> <ul style="list-style-type: none"> • AVG() • BIT_AND() • BIT_OR()

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		<ul style="list-style-type: none"> • BIT_XOR() • COUNT() • JSON_ARRAYAGG() • JSON_OBJECTAGG() • MAX() • MIN() • STDDEV_POP() • STDDEV() • STD() • STDDEV_SAMP() • SUM() • VAR_POP() • VARIANCE() • VAR_SAMP() <p>MySQL mendukung fungsi jendela yang, untuk setiap baris dari kueri, melakukan perhitungan menggunakan baris yang terkait dengan baris itu.</p> <ul style="list-style-type: none"> • CUME_DIST() • DENSE_RANK() • FIRST_VALUE() • LAG() • LAST_VALUE() • LEAD() • NTH_VALUE() • NTILE() • PERCENT_RANK() • RANK()

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		<ul style="list-style-type: none"> • ROW_NUMBER()
PostgreSQL	MySQL 8.x	Ditambahkan dukungan untuk kendala CHECK.
Semua	SQL Server	<p>SQL Server 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: Menambahkan dukungan untuk Graph Databases Node dan tipe tabel Edge.</p> <p>SQL Server 2016: Menambahkan dukungan untuk TABEL TEMPORAL.</p>
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.
PostgreSQL/Aurora PostgreSQL	Semua	<p>Ditambahkan tampilan sistem sysindex emulasi.</p> <p>Jika ada pernyataan SELECT dalam prosedur tanpa menentukan INTO, parameter INOUT p_refcur tipe refcursor dibuat untuk prosedur pada target.</p>

Rilis 1.0.640 Db2 perubahan LUW

Daftar tabel berikut membangun 1.0.640 perubahan di mana DB2 LUW adalah mesin sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
DB2 LUW	RDS MariaDB 10.4	Menambahkan dukungan RDS MariaDB 10.4 untuk semua vendor pemrosesan transaksional online (OLTP).

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Tingkatkan ke Amazon Corretto JDK 11 dari JDK 8. Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
DB2 LUW	MySQL 8.0.17	Ditambahkan dukungan kendala CHECK.
Semua	SQL Server	<p>SQL Server 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: Menambahkan dukungan untuk Graph Databases Node dan tipe tabel Edge.</p> <p>SQL Server 2016: Menambahkan dukungan untuk TABEL TEMPORAL.</p>
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.

Rilis 1.0.640 Perubahan Teradata

Tabel berikut mencantumkan build 1.0.640 perubahan mesin sumber Teradata.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	<p>Menambahkan dukungan untuk pernyataan MERGE dan QUALIFY.</p> <p>Dihapus LOCKING ROWS FOR ACCESS klausa dari pernyataan Teradata.</p> <p>Ditambahkan dukungan untuk fungsi CAST.</p>

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Tingkatkan ke Amazon Corretto JDK 11 dari JDK 8. Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Teradata	Teradata	Peningkatan yang diterapkan di REGEXP_INSTR () dan REGEXP_SUBSTR ().
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
Semua	SQL Server	<p>SQL Server 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: Menambahkan dukungan untuk Graph Databases Node dan tipe tabel Edge.</p> <p>SQL Server 2016: Menambahkan dukungan untuk TABEL TEMPORAL.</p>
Teradata	Semua	Ditambahkan dukungan untuk REGEXP_INSTR () dan REGEXP_SUBSTR ().
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.
Teradata	Amazon Redshift	<p>Menerapkan kemampuan untuk menyimpan SQL dari pohon sumber ke dalam satu file atau beberapa file secara bertahap menggunakan pengaturan di Pengaturan Proyek, Simpan sebagai SQL dan Terapkan, Daftar dropdown: File tunggal/Beberapa file.</p> <p>Peningkatan yang diterapkan dalam konversi tampilan dan prosedur.</p>
Teradata	Semua	Ditambahkan dukungan untuk Teradata versi 16.20

Rilis 1.0.640 perubahan untuk mesin lain

Tabel berikut mencantumkan perubahan build 1.0.640 untuk mesin sumber lainnya.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Sybase	RDS MariaDB 10.4	Menambahkan dukungan RDS MariaDB 10.4 untuk semua vendor pemrosesan transaksional online (OLTP).
SAP ASE	MariaDB	Diimplementasikan sebagai berikut: <ul style="list-style-type: none"> • MariaDB 10.4 • EKSEKUSI pernyataan LANGSUNG • Definisi DEFAULT • PERIKSA dukungan kendala
SAP ASE	PostgreSQL 12.x	Ditambahkan dukungan untuk kolom yang dihasilkan.
Semua	Semua	Tingkatkan ke Amazon Corretto JDK 11 dari JDK 8. Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
SAP ASE	MySQL 8.0.17	Ditambahkan dukungan kendala CHECK.
Semua	SQL Server	SQL Server 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE_FOR_SEQUENTIAL_KEY. SQL Server 2017: Menambahkan dukungan untuk Graph Databases Node dan tipe tabel Edge.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		SQL Server 2016: Menambahkan dukungan untuk TABEL TEMPORAL.
Vertica	Amazon Redshift	Ditambahkan dukungan untuk gaya distribusi = AUTO.
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.
Amazon Redshift	Amazon Redshift	Fungsi bawaan yang tidak didukung dalam pernyataan DML diganti dengan NULL sebagai placeholder.
Sybase	PostgreSQL	Menambahkan dukungan untuk fungsi asli.
SAP ASE	MySQL/Aurora MySQL	Tingkat isolasi default untuk Aurora MySQL adalah REPEATABLE READ. Menyetel tingkat isolasi GLOBAL tidak didukung di Aurora MySQL. Hanya lingkup sesi yang dapat diubah. Perilaku default transaksi adalah menggunakan REPEATABLE READ dan pembacaan yang konsisten. Aplikasi yang dirancang untuk berjalan dengan READ COMMITTED mungkin perlu dimodifikasi. Atau Anda dapat secara eksplisit mengubah default menjadi READ COMMITTED.
SAP ASE	PostgreSQL	Menambahkan dukungan untuk fungsi CONVERT (optimis) tanpa paket ekstensi.
SAP ASE	Semua	Ditambahkan tampilan sistem sysindex emulasi. Jika ada pernyataan SELECT dalam prosedur tanpa menentukan INTO, parameter INOUT p_refcur tipe refcursor dibuat untuk prosedur pada target.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Greenplum	Amazon Redshift	Diimplementasikan CREATE TEMPORARY TABLE sebagai berikut:

Riwayat dokumen

Tabel berikut menjelaskan perubahan penting pada AWS Schema Conversion Tool (AWS SCT) panduan pengguna setelah Januari 2018.

Untuk pemberitahuan tentang pembaruan dokumentasi ini, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
AWS SCT membangun #1 .0.672	Build 1.0.672 menyediakan dukungan Amazon RDS untuk PostgreSQL 15 sebagai target dan Microsoft SQL Server versi 2022 sebagai sumber. Ini juga menambahkan dukungan untuk fitur Amazon Redshift baru dalam kode yang dikonversi, mengimplementasikan beberapa peningkatan konversi untuk sumber IBM Db2 untuk z/OS, dan menyelesaikan sejumlah masalah konversi.	8 Mei 2023
AWS SCT membangun #1 .0.671	Build 1.0.671 memberikan dukungan migrasi dari Apache Oozie ke AWS Step Functions. Ini juga menambahkan dukungan untuk BigQuery sebagai sumber untuk proses penilaian multiserver. Selain itu, ia menambahkan pengaturan konversi baru untuk IBM Db2 untuk z/OS sebagai sumber	8 Maret 2023

dan menyelesaikan sejumlah masalah konversi.

[AWS SCT membangun
#1 .0.670](#)

Build 1.0.670 menyediakan dukungan migrasi dari Hadoop ke Amazon EMR. Ini juga menambahkan dukungan untuk Azure Synapse Analytics sebagai sumber untuk proses penilaian multiserver. Selain itu, ini meningkatkan konversi kode SQL yang tertanam dalam aplikasi Java dan menyelesaikan sejumlah masalah konversi.

Januari 23, 2023

[AWS SCT membangun
#1 .0.669](#)

Build 1.0.669 mengimplementasikan dukungan partisi asli untuk migrasi data dari gudang data Oracle. Ini juga meningkatkan proses penilaian multiserver, menambahkan fitur baru dalam agen ekstraksi data, dan menyelesaikan sejumlah masalah konversi.

Desember 19, 2022

[AWS SCT membangun
#1 .0.668](#)

Build 1.0.668 mengimple-
mentasikan partisi virtual
otomatis untuk migrasi data
dari database Greenplum
, dan menambahkan
dukungan migrasi data dari
database Snowflake ke
Amazon Redshift. Hal ini juga
meningkatkan konversi kode
SQL yang tertanam dalam C
aplikasi dan menyelesaikan
sejumlah masalah konversi.

November 16, 2022

[AWS SCT membangun
#1 .0.667](#)

Build 1.0.667 memberika
n dukungan untuk mesin
ekstrak, transformasi, dan
beban Informatica (ETL)
sebagai sumber migrasi. Ini
juga memperbarui versi paket
ekstensi, meningkatkan versi
driver minimum yang didukung
untuk Amazon Redshift, dan
menyelesaikan sejumlah
masalah konversi.

Oktober 13, 2022

[AWS SCT membangun
#1 .0.666](#)

Build 1.0.666 meningkatkan
konversi aplikasi Java dengan
menambahkan dukungan
untuk kerangka kerja. MyBatis
Ini juga menambahkan fungsi
baru ke dalam paket ekstensi,
meningkatkan loader metadata
sumber, dan menyelesaikan
sejumlah masalah konversi.

September 20, 2022

[AWS SCT membangun
#1 .0.665](#)

Build 1.0.665 menyediakan dukungan BigQuery sebagai sumber migrasi. Ini juga mengimplementasikan dukungan dari versi baru dari file konfigurasi fitur Babelfish . Selain itu, ini meningkatkan konversi gudang data ke Amazon Redshift, dan menyelesaikan sejumlah masalah konversi.

29 Agustus 2022

[AWS SCT membangun
#1 .0.664](#)

Build 1.0.664 memberikan dukungan Amazon Redshift Tanpa Server sebagai sumber atau target migrasi. Ini juga mengimplementasikan penyeimbangan memori otomatis dalam tugas penggalian data, dan memperbaiki kesalahan yang AWS SCT tidak dapat terhubung ke perangkat. AWS Snowball Selain itu, ia menambahkan kemampuan untuk mengubah pemeriksaan kolom dalam aturan migrasi, meningkatkan antarmuka pengguna, dan menyelesaikan sejumlah masalah konversi.

Juli 14, 2022

[AWS SCT membangun
#1 .0.663](#)

Build 1.0.663 menambahkan dukungan untuk Babelfish untuk Aurora PostgreSQL 1.2.0 dan meningkatkan kemampuan laporan penilaian multiserver. Ini juga menambahkan fitur baru dalam aturan migrasi, memperbaiki dua kesalahan loader, dan menyelesaikan sejumlah masalah konversi.

20 Juni 2022

[AWS SCT membangun
#1 .0.662](#)

Build 1.0.662 mengimplementasikan konversi kode SQL dalam aplikasi C # dan meningkatkan alur kerja laporan penilaian multiserver. Ini juga menambahkan beberapa perbaikan konversi dan menyelesaikan sejumlah masalah konversi.

Mei 19, 2022

[AWS SCT membangun
#1 .0.661](#)

Membangun 1.0.661 menyediakan dukungan IBM Db2 untuk z/OS sebagai sumber migrasi. Ini juga menambahkan dukungan untuk konversi skrip ekstrak, transformasi, dan beban (ETL) ke AWS Glue Studio dan menyelesaikan sejumlah masalah konversi.

April 21, 2022

[AWS SCT membangun
#1 .0.660](#)

Build 1.0.660 menyediakan dukungan PostgreSQL mayor versi 14 dan MariaDB 10.6 sebagai target migrasi. Ini juga menambahkan dukungan untuk konversi indeks Oracle untuk tampilan terwujud, dan menyelesaikan sejumlah masalah konversi.

21 Maret 2022

[AWS SCT membangun
#1 .0.659](#)

Build 1.0.659 menyediakan dukungan PostgreSQL mayor versi 13 pada Aurora PostgreSQL kompatibel Edition sebagai target migrasi. Ini mengimplementasikan konversi kode SQL di C# aplikasi, menambahkan dukungan dari Oracle Unified Auditing, dan menyelesaikan sejumlah masalah konversi.

Februari 21, 2022

[AWS SCT membangun
#1 .0.658](#)

Build 1.0.658 menyediakan integrasi dengan AWS Secrets Manager dan menambahkan dukungan platform database target virtual Amazon Redshift. Ini juga menambahkan sejumlah perbaikan konversi dan perbaikan bug.

Januari 20, 2022

[AWS SCTmembangun
#1 .0.657](#)

Build 1.0.657 meningkatkan konversi dari Microsoft SQL Server ke Edisi yang kompatibel dengan Aurora PostgreSQL, Amazon RDS untuk PostgreSQL, dan tujuan migrasi lainnya. Ini juga menambahkan sejumlah peningkatan antarmuka pengguna dan perbaikan bug.

Desember 20, 2021

[AWS SCTmembangun
#1 .0.656](#)

Build 1.0.656 menyediakan dukungan dari beberapa database sumber dan target dalam satu proyek. Ini juga menambahkan konversi, strategi optimasi, dan perbaikan umum dan sejumlah perbaikan bug.

November 22, 2021

[AWS SCTmembangun
#1 .0.655](#)

Build 1.0.655 mengimplementasikan konversi skrip FastExport pekerjaan Teradata ke Amazon Redshift RSQL dan meningkatkan versi driver minimum yang didukung untuk Greenplum menjadi 42.2.5. Ini juga menambahkan sejumlah perbaikan dan perbaikan bug.

Oktober 18, 2021

[AWS SCTmembangun
#1 .0.654](#)

Build 1.0.654 mengimple-
mentasikan konversi skrip
Shell, TeradataFastLoad,
dan Teradata Basic Teradata
Query (BTEQ) ke Amazon
Redshift RSQL. Ini juga
menyelesaikan sejumlah
masalah konversi dan
menambahkan sejumlah
perbaikan dan perbaikan bug.

September 16, 2021

[AWS SCTmembangun
#1 .0.653](#)

Build 1.0.653 mengimple-
mentasikan konversi SQL
dinamis yang dibuat dalam
fungsi atau prosedur yang
disebut. Ini juga meningkatkan
konversi rutinitas terenkripsi
dan menambahkan sejumlah
perbaikan dan perbaikan bug.

Agustus 10, 2021

[AWS SCTmembangun
#1 .0.652](#)

Build 1.0.652 mengimple-
mentasikan mode perintah
skrip di antarmuka baris
perintah dan menerapkan
aturan optimasi skema. Ini
juga menambahkan sejumlah
peningkatan konversi dan
kinerja dan perbaikan bug.

30 Juni 2021

[AWS SCTmembangun
#1 .0.651](#)

Build 1.0.651 menambahk-
an sejumlah perbaikan dan
perbaikan bug. Ini juga
menyediakan akses ke salinan
awal Referensi AWS Schema
Conversion Tool CLI.

4 Juni 2021

[AWS SCT membangun
#1 .0.650](#)

Build 1.0.650 mengimple-
mentasikan dukungan Amazon
RDS untuk PostgreSQL 13
sebagai database target,
memperbarui agen ekstrakto-
r. Ini juga meningkatkan versi
driver JDBC minimum yang
didukung untuk Microsoft
SQL Server, Azure, dan
Azure Synapse. Selain itu,
ia menambahkan sejumlah
peningkatan konversi dan
perbaikan bug.

30 April 2021

[AWS SCT membangun
#1 .0.649](#)

Build 1.0.649 mengimple-
mentasikan dukungan
MariaDB 10.5 sebagai
database target dan
mengimplementasikan
penyempurnaan fungsi untuk
konversi fungsi bawaan
Oracle. Ini juga menambahkan
sejumlah peningkatan konversi
dan kinerja dan perbaikan bug.

29 Maret 2021

[AWS SCT membangun
#1 .0.648](#)

Build 1.0.648 menambahkan
sejumlah perbaikan konversi
dan perbaikan bug.

22 Februari 2021

[AWS SCTmembangun
#1 .0.647](#)

Build 1.0.647 menambahkan dukungan fitur Database Mail di Amazon RDS, mengimple mentasikan pemuatan dan konversi komentar pada objek penyimpanan. Hal ini juga menambahkan AWS SCT Data Migration Service Assessor dan AWS SCT Wizard dan mengimple mentasikan antarmuka pengguna filter pohon. Selain itu, ia menambahkan bagian yang didesain ulang dalam Laporan Penilaian dan sejumlah peningkatan dan perbaikan bug.

15 Januari 2021

[AWS SCTmembangun
#1 .0.646](#)

Build 1.0.646 menambahk an dukungan untuk tipe data INTERVAL, kolom Identitas , dan konversi kursor, dan menambahkan sejumlah perbaikan dan perbaikan bug.

28 Desember 2020

[AWS SCTmembangun
#1 .0.645](#)

Build 1.0.645 menambahkan dukungan untuk ETL SSIS untuk AWS Glue konversi dan sejumlah perbaikan dan perbaikan bug.

16 November 2020

[AWS SCTmembangun
#1 .0.643-1.0.644](#)

Build 1.0.644 menambahk an sejumlah peningkat an konversi, kinerja, dan antarmuka pengguna serta perbaikan bug.

Oktober 14, 2020

AWS SCTmembangun #1 .0.642	Membangun 1.0.642 mengimplementasikan konversi paket ETL dari Microsoft SQL Server Integrati on Services AWS Glue dan menambahkan sejumlah perbaikan dan perbaikan bug.	28 Agustus 2020
AWS SCTmembangun #1 .0.641	Menambahkan dukungan SSL untuk ekstraktor data. Build juga mencakup sejumlah perbaikan dan perbaikan.	17 Juli 2020
AWS SCTmembangun #1 .0.633-1.0.640	Diupgrade dari JDK 8 ke Amazon Corretto JDK 11. Ditambahkan tabel mengident ifikasi upgrade lainnya, perubahan, dan perbaikan.	22 Juni 2020
Ketersediaan AWS WQF	AWS SCTtidak lagi menyediak an alat AWS Workload Qualification Framework (AWS WQF) untuk diunduh.	19 Juni 2020
AWS SCTmembangun #1 .0.632	SCT UI - Menambahkan tab baru untuk menampilkan kesalahan yang terjadi saat menerapkan skrip. Anda sekarang dapat menyimpan pohon sumber sebagai SQL saat mengonversi dari SAP ASE. Perbaikan untuk konversi ke PostgreSQL atau Aurora PostgreSQL atau Redshift.	19 November 2019

[AWS SCT membangun #1 .0.631 dan #1 .0.630 \(gabungan\)](#)

Dukungan Rowids yang lebih baik di Oracle, dan untuk objek sistem di Microsoft SQL Server dan SAP ASE. Penanganan yang lebih baik untuk penentu skema SQL Server yang hilang. Dukungan yang lebih baik untuk konversi dari Greenplum ke Redshift. Peningkatan dukungan untuk konversi kode yang disimpan saat pindah ke Amazon Redshift, MariaDB, MySQL, dan PostgreSQL.

30 September 2019

[AWS SCT membangun #1 .0.629](#)

Dukungan untuk prosedur yang disimpan untuk konversi dari Netezza. Peningkatan dukungan untuk konversi ke Amazon Redshift, DynamoDB, MySQL, dan PostgreSQL. Ditambahkan dukungan untuk SAP ASE 12.5 sebagai sumber.

20 Agustus 2019

[AWS SCT membangun #1 .0.628](#)

Dukungan untuk emulasi layanan untuk konversi dari DB2, SQL Server dan Oracle. Penyempurnaan untuk konversi ke Amazon Redshift, termasuk lebih banyak dukungan untuk kursor dan prosedur yang disimpan.

22 Juni 2019

AWS SCTmembangun #1 .0.627	Dukungan untuk konversi dari SQL Server ke prosedur tersimpan di Amazon Redshift. Perangkat tambahan untuk mengkonversi ke PostgreSQL 11 dan MySQL 8.0.	31 Mei 2019
AWS SCTmembangun #1 .0.626	PostgreSQL 11 dan MySQL 8.0 sekarang didukung sebagai target. SAP ASE 15.5 sekarang didukung sebagai sumber.	26 April 2019
AWS SCTmembangun #1 .0.625	Pembaruan mencakup kemampuan untuk mengonversi Teradata BTEQ keAWS Glue, dukungan untuk konversi ke MariaDB 10.3 dengan dukungan mode kompatibilitas Oracle, dukungan untuk SAP ASE 15.7, dan substitusi layanan untuk meniru fungsionalitas yang hilang.	25 Maret 2019

[AWS SCTmembangun
#1 .0.624](#)

Pembaruan mencakup kemampuan untuk mengonversi Oracle ETL keAWS Glue, dan dukungan untuk konversi dari Microsoft SQL Server, Oracle, dan IBM Db2 LUW ke Amazon RDS untuk MariaDB. Kami juga menambahkan dukungan untuk konversi dari SAP ASE ke RDS untuk MySQL dan Amazon Aurora dengan kompatibilitas MySQL. Selain itu, kami menambahkan dukungan untuk ekstensi Orafce selama konversi Oracle ke PostgreSQL.

22 Februari 2019

[AWS SCTmembangun
#1 .0.623](#)

Pembaruan mencakup kemampuan untuk mengonversi database SAP ASE, dan kemampuan untuk mengonversi skrip T-SQL, DML, dan DDL ke kode atau komponen yang setara. Kami juga menambahkan emulasi Oracle dan Microsoft SQL Server untuk meningkatkan konversi.

25 Januari 2019

[AWS SCTmembangun
#1 .0.622](#)

Pembaruan mencakup Kerangka Kerja Kualifikasi Beban Kerja, yang menganalisis beban kerja untuk seluruh migrasi, termasuk modifikasi database dan aplikasi.

20 Desember 2018

AWS SCTmembangun #1 .0.621	Pembaruan mencakup dukungan untuk Aurora PostgreSQL 10 sebagai target, dan kemampuan untuk bermigrasi dari Netezza menggunakan opsi tabel eksternal.	21 November 2018
AWS SCTmembangun #1 .0.620	Pembaruan mencakup kemampuan untuk menyimpan skrip SQL, dan dukungan untuk kursor global Oracle saat bermigrasi ke MySQL.	22 Oktober 2018
AWS SCTmembangun #1 .0.619	Pembaruan mencakup dukungan untuk migrasi dari Apache Cassandra ke DynamoDB, dan dukungan untuk Vertica 9 sebagai sumber.	20 September 2018
AWS SCTmembangun #1 .0.618	Pembaruan mencakup laporan penilaian yang diperluas, dukungan untuk mengonversi Oracle Rowids, dan dukungan untuk tabel yang ditentukan pengguna SQL Server.	24 Agustus 2018
AWS SCTmembangun #1 .0.617	Pembaruan mencakup laporan penilaian yang diperluas, dukungan untuk mengonversi Oracle Rowids, dan dukungan untuk tabel yang ditentukan pengguna SQL Server.	24 Juli 2018

AWS SCTmembangun #1 .0.616	Pembaruan mencakup dukungan untuk RDS saat mengonversi dari Oracle ke Amazon RDS untuk Oracle, mengonversi objek jadwal Oracle, dan dukungan untuk pekerjaan Oracle. Partisi, dan Db2 LUW versi 10.1.	26 Juni 2018
AWS SCTmembangun #1 .0.615	Pembaruan mencakup dukungan untuk pernyataan SQL Server ke PostgreSQL GOTO, partisi PostgreSQL 10, dan Db2 LUW versi 10.1.	24 Mei 2018
AWS SCTmembangun #1 .0.614	Pembaruan mencakup dukungan untuk Oracle ke Oracle DB Links, SQL Server ke fungsi inline PostgreSQL, dan emulasi objek sistem Oracle.	25 April 2018
AWS SCTmembangun #1 .0.613	Pembaruan termasuk dukungan untuk Db2 LUW, konversi file SQL* Plus, dan SQL Server Windows Authentication.	28 Maret 2018
AWS SCTmembangun #1 .0.612	Pembaruan mencakup dukungan untuk pemetaan tipe data kustom, perbandingan skema untuk Oracle 10, dan konversi Oracle ke PostgreSQL variabel global.	22 Februari 2018

[AWS SCT membangun #1 .0.611](#)

Pembaruan termasuk dukungan untuk Oracle untuk pernyataan dinamis PostgreSQL, membuka file log dengan memilih pesan kesalahan, dan kemampuan untuk menyembunyikan skema dalam tampilan pohon.

23 Januari 2018

Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting pada AWS Schema Conversion Tool (AWS SCT) panduan pengguna sebelum Januari 2018.

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.608	Dukungan titik akhir FIPS untuk Amazon S3	Anda sekarang dapat meminta AWS SCT untuk terhubung ke Amazon S3 dan Amazon Redshift dengan menggunakan titik akhir FIPS untuk mematuhi persyaratan keamanan Standar Pemrosesan Informasi Federal. Untuk informasi selengkapnya, lihat Menyimpan AWS kredensi .	17 November 2017
1.0.607	Dukungan titik akhir FIPS untuk Amazon S3	Anda sekarang dapat meminta AWS SCT untuk terhubung ke Amazon S3 dan Amazon Redshift dengan menggunakan titik akhir FIPS untuk mematuhi persyaratan keamanan Standar Pemrosesan Informasi Federal. Untuk informasi selengkapnya, lihat Menyimpan AWS kredensi .	Oktober 30, 2017
1.0.607	Tugas ekstraksi data dapat	Saat Anda membuat tugas ekstraksi data, Anda sekarang dapat memilih untuk mengabaikan objek besar (LOB) untuk	Oktober 30, 2017

Versi	Perubahan	Deskripsi	Tanggal diubah
	mengabaikan LOB	mengurangi jumlah data yang Anda ekstrak. Untuk informasi selengkapnya, lihat Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data .	
1.0.605	Akses log tugas agen ekstraksi data	Anda sekarang dapat mengakses log tugas agen ekstraksi data dari tautan yang nyaman di antarmuka AWS Schema Conversion Tool pengguna. Untuk informasi selengkapnya, lihat Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data .	28 Agustus 2017
1.0.604	Penyempurnaan konverter	AWS Schema Conversion ToolMesin telah ditingkatkan untuk menawarkan konversi yang lebih baik untuk migrasi heterogen.	24 Juni 2017
1.0.603	Agen ekstraksi data mendukung filter	Anda sekarang dapat memfilter data yang diambil agen ekstraksi dari gudang data Anda. Untuk informasi selengkapnya, lihat Membuat aturan migrasi data di AWS SCT .	16 Juni 2017
1.0.603	AWS SCTmendukung versi gudang data tambahan	Sekarang Anda dapat menggunakan skema AWS Schema Conversion Tool untuk mengonversi skema Teradata 13 dan Oracle Data Warehouse 10 ke skema Amazon Redshift yang setara. Untuk informasi selengkapnya, lihat Mengonversi skema gudang data ke Amazon Redshift menggunakan AWS SCT .	16 Juni 2017

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.602	Agen ekstraksi data mendukung gudang data tambahan	Anda sekarang dapat menggunakan agen ekstraksi data untuk mengekstrak data dari gudang data Microsoft SQL Server Anda. Untuk informasi selengkapnya, lihat Memigrasi data dari gudang data lokal ke Amazon Redshift .	11 Mei 2017
1.0.602	Agen ekstraksi data dapat menyalin data ke Amazon Redshift	Agen ekstraksi data sekarang memiliki tiga mode unggahan. Anda sekarang dapat menentukan apakah akan hanya mengekstrak data Anda, untuk mengekstrak data Anda dan hanya mengunggahnya ke Amazon S3, atau untuk mengekstrak, mengunggah, dan menyalin data Anda langsung ke Amazon Redshift. Untuk informasi selengkapnya, lihat Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data .	11 Mei 2017
1.0.601	AWS SCT mendukung gudang data tambahan	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengonversi skema Vertica dan Microsoft SQL Server Anda ke skema Amazon Redshift yang setara. Untuk informasi selengkapnya, lihat Mengonversi skema gudang data ke Amazon Redshift menggunakan AWS SCT .	April 18, 2017
1.0.601	Agen ekstraksi data mendukung gudang data tambahan	Anda sekarang dapat menggunakan agen ekstraksi data untuk mengekstrak data dari gudang data Greenplum, Netezza, dan Vertica Anda. Untuk informasi selengkapnya, lihat Memigrasi data dari gudang data lokal ke Amazon Redshift .	April 18, 2017

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.601	Agen ekstraksi data mendukung sistem operasi tambahan	Anda sekarang dapat menginstal agen ekstraksi data di komputer yang menjalankan sistem operasi macOS dan Microsoft Windows. Untuk informasi selengkapnya, lihat Instalasi agen ekstraksi .	April 18, 2017
1.0.601	Agen ekstraksi data mengunggah ke Amazon S3 secara otomatis	Agen ekstraksi data sekarang mengunggah data yang Anda ekstrak ke Amazon S3 secara otomatis. Untuk informasi selengkapnya, lihat Output tugas ekstraksi data .	April 18, 2017
1.0.600	Agen Ekstraksi Data	Anda sekarang dapat menginstal agen ekstraksi data yang mengekstrak data dari gudang data Anda dan menyiapkannya untuk digunakan dengan Amazon Redshift. Anda dapat menggunakan AWS Schema Conversion Tool untuk mendaftarkan agen dan membuat tugas ekstraksi data untuk mereka. Untuk informasi selengkapnya, lihat Memigrasi data dari gudang data lokal ke Amazon Redshift .	Februari 16, 2017
1.0.600	Umpan Balik Pelanggan	Anda sekarang dapat memberikan umpan balik tentang AWS Schema Conversion Tool. Anda dapat mengajukan laporan bug, Anda dapat mengirimkan permintaan fitur, atau Anda dapat memberikan informasi umum. Untuk informasi selengkapnya, lihat Memberikan umpan balik .	Februari 16, 2017

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.502	Integrasi dengan AWS DMS	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk membuat AWS DMS titik akhir dan tugas. Anda dapat menjalankan dan memantau tugas-tugas dari AWS SCT. Untuk informasi selengkapnya, lihat Menggunakan AWS SCT dengan AWS DMS .	20 Desember 2016
1.0.502	Amazon Aurora dengan kompatibilitas PostgreSQL sebagai database target	AWS Schema Conversion Tool sekarang mendukung Amazon Aurora dengan kompatibilitas PostgreSQL sebagai database target. Untuk informasi selengkapnya, lihat Mengonversi skema database menggunakan AWS SCT .	20 Desember 2016
1.0.502	Dukungan untuk profil	Anda sekarang dapat menyimpan profil yang berbeda di AWS Schema Conversion Tool dan dengan mudah beralih di antara mereka. Untuk informasi selengkapnya, lihat Menyimpan profil AWS layanan di AWS SCT .	20 Desember 2016
1.0.501	Dukungan untuk Greenplum Database dan Netezza	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengonversi skema gudang data Anda dari Greenplum Database dan Netezza ke Amazon Redshift. Untuk informasi selengkapnya, lihat Mengonversi skema gudang data ke Amazon Redshift menggunakan AWS SCT .	17 November 2016

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.501	Pengoptimalkan pergeseran merah	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengoptimalkan database Amazon Redshift Anda. Untuk informasi selengkapnya, lihat Mengoptimalkan Amazon Redshift dengan menggunakan AWS SCT .	17 November 2016
1.0.500	Aturan pemetaan	Sebelum Anda mengonversi skema Anda dengan AWS Schema Conversion Tool, Anda sekarang dapat mengatur aturan yang mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek. Untuk informasi selengkapnya, lihat Membuat aturan migrasi di AWS SCT .	Oktober 4, 2016
1.0.500	Pindah ke cloud	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk menyalin skema database lokal yang ada ke instans DB Amazon RDS yang menjalankan mesin yang sama. Anda dapat menggunakan fitur ini untuk menganalisis potensi penghematan biaya pindah ke cloud dan mengubah jenis lisensi Anda. Untuk informasi selengkapnya, lihat Membuat laporan penilaian migrasi dengan AWS SCT .	Oktober 4, 2016
1.0.400	Konversi skema gudang data	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengonversi skema gudang data Anda dari Oracle dan Teradata ke Amazon Redshift. Untuk informasi selengkapnya, lihat Mengonversi skema gudang data ke Amazon Redshift menggunakan AWS SCT .	13 Juli 2016

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.400	Konversi SQL aplikasi	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengkonversi SQL di C ++, C #, Java, atau kode aplikasi lainnya. Untuk informasi selengkapnya, lihat Mengonversi aplikasi SQL menggunakan AWS SCT .	13 Juli 2016
1.0.400	Fitur baru	AWS Schema Conversion Tool Sekarang berisi paket ekstensi dan wizard untuk membantu Anda menginstal, membuat, dan mengkonfigurasi AWS Lambda fungsi dan pustaka Python untuk menyediakan email, penjadwalan pekerjaan, dan fitur lainnya. Untuk informasi lebih lanjut, lihat Menggunakan AWS Lambda fungsi dari paket AWS SCT ekstensi dan Menggunakan pustaka khusus untuk paket AWS SCT ekstensi .	13 Juli 2016
1.0.301	Dukungan SSL	Anda sekarang dapat menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database sumber Anda saat Anda menggunakan file. AWS Schema Conversion Tool	19 Mei 2016
1.0.203	Fitur baru	Menambahkan dukungan untuk MySQL dan PostgreSQL sebagai database sumber untuk konversi.	11 April 2016

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.202	Rilis pemeliharaan	Menambahkan dukungan untuk mengedit SQL dikonversi yang dihasilkan untuk mesin database target. Menambahkan kemampuan seleksi yang ditingkatkan dalam database sumber dan menargetkan tampilan pohon contoh DB. Menambahkan dukungan untuk menghubungkan ke database sumber Oracle menggunakan nama Transparent Network Substrate (TNS).	2 Maret 2016
1.0.200	Rilis pemeliharaan	Menambahkan dukungan untuk PostgreSQL sebagai mesin database target. Menambahkan kemampuan untuk menghasilkan skema dikonversi sebagai script dan untuk menyimpan script ke file sebelum menerapkan skema untuk contoh DB target.	14 Januari 2016
1.0.103	Rilis pemeliharaan	Menambahkan kemampuan proyek offline, kemampuan untuk memeriksa versi baru, dan manajemen memori dan kinerja.	Desember 2, 2015
1.0.101	Rilis pemeliharaan	Menambahkan wizard Create New Database Migration Project. Menambahkan kemampuan untuk menyimpan laporan penilaian migrasi database sebagai file PDF.	19 Oktober 2015
1.0.100	Rilis pratinjau	Menyediakan panduan pengguna untuk rilis AWS Schema Conversion Tool pratinjau.	7 Oktober 2015

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.