



Guida per l'utente

AWS Schema Conversion Tool



Version 1.0.672

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

AWS Schema Conversion Tool: Guida per l'utente

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

I marchi e il trade dress di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in qualsiasi modo che possa causare confusione tra i clienti o in qualsiasi modo che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Cos'è AWS SCT?	1
Panoramica della conversione dello schema	5
Dare feedback	6
Installazione, verifica e aggiornamento	8
Installazione AWS SCT	8
Installazione delle versioni precedenti	9
Verifica del download del AWS SCT file	10
Verifica del checksum del file AWS SCT	10
Verifica dei file AWS SCT RPM su Fedora	11
Verifica dei file AWS SCT DEB su Ubuntu	12
Verifica del file AWS SCT MSI su Microsoft Windows	12
Scaricamento dei driver di database richiesti	13
Installazione dei driver JDBC su Linux	16
Memorizzazione dei percorsi dei driver nelle impostazioni globali	17
Aggiornamento AWS SCT	18
AWS SCT CLI	19
Utilizzo dell'interfacciaAWS SCT utente	20
La finestra del progetto	20
Avvio di AWS SCT	22
Creazione di un progetto	22
Utilizzo di una nuova procedura guidata di progetto	23
Salvare e aprire un progetto	26
Aggiungere un server	27
Utilizzo di una modalità offline	28
Utilizzo dei filtri ad albero	29
.....	30
Importazione di un elenco di file per il filtro ad albero	32
Nascondere schemi	33
Gestione del rapporto di valutazione della migrazione del database	35
Conversione dello schema	39
Applicazione del codice convertito	42
MemorizzazioneAWS dei profili	43
ArchiviazioneAWS delle credenziali	44
Impostazione del profilo predefinito per un progetto	46

Autorizzazioni per l'utilizzo del profiloAWS di servizio	46
Utilizzo di AWS Secrets Manager	47
Memorizzare le password del database	48
Utilizzo della visualizzazione Union All per progetti con tabelle partizionate	48
Tasti di scelta rapida	49
Nozioni di base	51
Origini per AWS SCT	53
Crittografia delle connessioni Amazon RDS	54
Usare Apache Cassandra come fonte	57
Connessione ad Apache Cassandra come fonte	57
Usare Apache Hadoop come fonte	59
Prerequisiti per l'utilizzo di Apache Hadoop come sorgente	59
Autorizzazioni per Hive come fonte	60
Autorizzazioni per HDFS come fonte	61
Autorizzazioni per HDFS come destinazione	61
Connessione ad Apache Hadoop come fonte	61
Connessione a Hive e HDFS	63
Connessione ad Amazon EMR come destinazione	66
Usare Apache Oozie come fonte	69
Prerequisiti	70
Connessione ad Apache Oozie come sorgente	70
Autorizzazioni per AWS Lambda	72
Connessione a AWS Step Functions come obiettivo	74
Usare il database SQL di Azure come origine	75
Privilegi per il database SQL di Azure	76
Connessione al database SQL di Azure come origine	76
Utilizzo di IBM Db2 for z/OS come fonte	77
Prerequisiti per Db2 for z/OS	78
Privilegi per Db2 for z/OS	78
Connessione a Db2 for z/OS come sorgente	80
Privilegi per MySQL come obiettivo	82
Privilegi per PostgreSQL come obiettivo	83
Impostazioni di conversione da Db2 per z/OS a PostgreSQL	84
Utilizzo di IBM Db2 LUW come origine	86
Privilegi per Db2 LUW	86
Connessione a Db2 LUW come sorgente	88

Da Db2 LUW a PostgreSQL	90
Da Db2 LUW a MySQL	93
Utilizzo di MySQL come origine	94
Privilegi per MySQL	95
Connessione a MySQL come sorgente	95
Privilegi per PostgreSQL come destinazione	98
Utilizzo del database Oracle come fonte	98
Privilegi per Oracle	99
Connessione a Oracle come fonte	99
Da Oracle a PostgreSQL	103
Da Oracle a MySQL	110
Da Oracle ad Amazon RDS per Oracle	120
Utilizzo di PostgreSQL come origine	127
Privilegi per PostgreSQL	127
Connessione a PostgreSQL come sorgente	128
Privilegi per MySQL come obiettivo	130
Utilizzo di SAP ASE (Sybase ASE) come fonte	131
Privilegi per SAP ASE	132
Connessione a SAP ASE come sorgente	133
Privilegi per MySQL come obiettivo	135
Impostazioni di conversione da SAP ASE a MySQL	137
Privilegi per PostgreSQL come destinazione	137
Impostazioni di conversione da SAP ASE a PostgreSQL	138
Utilizzo di SQL Server come origine	139
Privilegi per Microsoft SQL Server	140
Utilizzo dell'autenticazione di Windows con Microsoft SQL Server	141
Connessione a SQL Server come origine	143
Da SQL Server a MySQL	146
Conversione da SQL Server a PostgreSQL	150
Da SQL Server ad Amazon RDS SQL Server	187
Fonti di data warehouse per AWS SCT	188
Utilizzo di Amazon Redshift come fonte	189
Usare Azure Synapse Analytics come fonte	194
Utilizzo di BigQuery come origine	200
Utilizzo del database Greenplum come fonte	206
Usare Netezza come fonte	212

Utilizzo di Oracle Data Warehouse come origine	222
Usare Snowflake come fonte	230
Utilizzo di SQL Server Data Warehouse come origine	239
Utilizzo di Teradata come fonte	246
Usare Vertica come fonte	261
Creazione di regole di mappatura	268
Nuova regola	269
Gestione delle regole	269
Obiettivi virtuali	271
Restrizioni	272
Creazione di report di conversione	273
Rapporti di valutazione della migrazione	273
Creazione di un rapporto di valutazione della migrazione del database	274
Visualizzazione del rapporto di valutazione	275
Salvataggio del rapporto di valutazione	279
Configurazione del rapporto di valutazione	281
Creazione di un rapporto di valutazione multiserver	285
Conversione di schemi di database	294
Creazione di regole di migrazione	296
Creazione di regole di migrazione	297
Esportazione delle norme sulla migrazione	299
Conversione dello schema	299
Conversione dello schema	300
Modifica dello schema convertito	302
Cancellare uno schema convertito	303
Gestione delle conversioni manuali	303
Modifica dello schema di origine	303
Modifica dello schema di destinazione	303
Aggiornamento e aggiornamento dello schema convertito	304
Salvare e applicare lo schema	305
Salvare lo schema convertito	305
Applicazione dello schema convertito	306
Lo schema del pacchetto di estensione	307
Confronto di schemi	307
Oggetti trasformati correlati	309
Conversione di schemi di data warehouse in Amazon Redshift	310

Autorizzazioni per Amazon Redshift	311
Scelta di strategie e regole di ottimizzazione	313
Raccolta o caricamento di statistiche	314
Creazione di regole di migrazione	316
Creazione di regole di migrazione	317
Esportazione delle norme sulla migrazione	318
Conversione dello schema	319
Conversione dello schema	319
Modifica dello schema convertito	322
Cancellare uno schema convertito	323
Gestione e personalizzazione delle chiavi	323
Argomenti correlati	324
Creazione e utilizzo del rapporto di valutazione	324
Creazione di un rapporto di valutazione della migrazione del database	324
Riepilogo	325
Oggetti d'azione	327
Salvataggio del rapporto di valutazione	327
Gestione delle conversioni manuali	328
Modifica dello schema di origine	329
Modifica dello schema di destinazione	329
Aggiornamento e aggiornamento dello schema convertito	330
Salvataggio e applicazione dello schema convertito	330
Salvare lo schema convertito in un file	331
Applicazione dello schema convertito	331
Lo schema del pacchetto di estensione	332
Librerie Python	333
Ottimizzazione di Amazon Redshift	333
Ottimizzazione del database Amazon Redshift	333
Conversione dei processi ETL	336
Conversione dei processi ETL inAWS Glue	337
Prerequisiti	338
Catalogo dati di AWS Glue	339
Restrizioni	339
Fase 1: creazione di un nuovo progetto	341
Fase 2: Creare unAWS Glueoccupazione	342
Conversione dei processi ETL utilizzando l'API Python perAWS Glue	343

Fase 1: crea un database	344
Fase 2: Creare una connessione	344
Fase 3: Creare unAWS Gluecingolato	346
Conversione di script ETL di Informatica	348
Conversione di SSIS inAWS Glue	353
Componenti SSIS supportati	357
Conversione di SSIS inAWS Glue Studio	359
Prerequisiti	359
Aggiungere pacchetti SSIS al tuoAWS SCTprogetto	361
Conversione dei pacchetti SSIS	362
CreandoAWS Glue Studiolavori	363
Creazione di un rapporto di valutazione della conversione SSIS	365
Componenti SSIS supportati	366
Conversione di Teradata BTEQ in Amazon Redshift RSQL	367
Aggiungere script BTEQ al tuoAWS SCTprogetto	369
Configurazione delle variabili di sostituzione negli script BTEQ	370
Conversione di script BTEQ	370
Gestione degli script BTEQ	371
Creazione di un rapporto di valutazione della conversione degli script BTEQ	372
Modifica e salvataggio degli script BTEQ convertiti	373
Conversione di script di shell in Amazon Redshift RSQL	373
Aggiungere script di shell al tuoAWS SCTprogetto	374
Configurazione delle variabili di sostituzione negli script di shell	375
Conversione degli script di shell	376
Gestione degli script di shell	376
Creazione di un rapporto di valutazione della conversione degli script di shell	377
Modifica e salvataggio degli script di shell convertiti	378
Conversione di TeradataFastExportad Amazon Redshift RSQL	378
AggiungendoFastExportscript di lavoro per i tuoiAWS SCTprogetto	379
Configurazione delle variabili di sostituzione inFastExportscript di lavoro	380
ConversioneFastExportscript di lavoro	381
GestireFastExportscript di lavoro	382
Creare unFastExporthrapporto di valutazione della conversione degli script di lavoro	383
Modifica e salvataggio dei file convertitiFastExportscript di lavoro	384
Conversione di TeradataFastLoadad Amazon Redshift RSQL	384
AggiungendoFastLoadscript di lavoro per i tuoiAWS SCTprogetto	385

Configurazione delle variabili di sostituzione inFastLoadscript di lavoro	386
ConversioneFastLoadscript di lavoro	387
GestireFastLoadscript di lavoro	388
Creare unFastLoadrapporto di valutazione della conversione degli script di lavoro	389
Modifica e salvataggio dei file convertitiFastLoadscript di lavoro	390
Conversione di TeradataMultiLoadad Amazon Redshift RSQL	390
AggiungendoMultiLoadscript di lavoro per i tuoiAWS SCTprogetto	391
Configurazione delle variabili di sostituzione inMultiLoadscript di lavoro	392
ConversioneMultiLoadscript di lavoro	393
GestireMultiLoadscript di lavoro	394
Creare unMultiLoadrapporto di valutazione della conversione degli script di lavoro	395
Modifica e salvataggio dei file convertitiMultiLoadscript di lavoro	396
Migrazione di framework per big data	397
Migrazione di Apache Hadoop su Amazon EMR	397
Panoramica	398
Passaggio 1: connessi ai tuoi cluster Hadoop	399
Fase 2: Impostazione delle regole di mappatura	399
Fase 3: Creazione di un rapporto di valutazione	401
Fase 4: Migra il tuo cluster Apache Hadoop su Amazon EMR	402
Esecuzione dello script CLI	403
Gestire il progetto di migrazione	404
Convertire Apache Oozie in AWS Step Functions	406
Panoramica	406
Passaggio 1: connessi ai tuoi servizi di origine e destinazione	408
Fase 2: Impostazione delle regole di mappatura	408
Fase 3: Configurazione dei parametri	409
Fase 4: Creazione di un rapporto di valutazione	411
Passaggio 5: converti i tuoi flussi di lavoro Apache Oozie in AWS Step Functions	412
Esecuzione dello script CLI	415
Nodi supportati	415
Uso di AWS SCT con AWS DMS	417
Utilizzo di unAWS SCTagente di replica conAWS DMS	417
Utilizzo di unAWS SCTagente di estrazione dati conAWS DMS	417
Aumento dei livelli di registrazione durante l'utilizzoAWS SCTconAWS DMS	417
Migrazione da un data warehouse ad Amazon Redshift	419
Prerequisiti	421

Impostazioni Amazon S3	422
Assumendo ruoli IAM	423
Impostazioni di sicurezza	424
Impostazioni di configurazione	425
Installazione degli agenti	425
Configurazione degli agenti	427
Installazione e configurazione di agenti di copia dedicati	429
Agenti di partenza	431
Registrazione degli agenti	432
Nascondere e recuperare informazioni per un agente AWS SCT	432
Creazione di regole per la migrazione dei dati	434
Modifica delle impostazioni di estrazione e copia per la migrazione dei dati	435
Ordinamento dei dati	438
Creazione, esecuzione e monitoraggio di qualsiasi AWS SCT attività	440
Esportazione e importazione di un'attività di estrazione dei dati	444
Estrazione dei dati tramite un dispositivo AWS Snowball Edge	445
tep-by-step Procedure S per la migrazione dei dati utilizzando AWS SCT Edge AWS Snowball	446
Output dell'attività di estrazione dei dati	449
Utilizzo del partizionamento virtuale	450
Limiti durante la creazione di partizionamenti virtuali	451
Tipo di partizione RANGE	451
Tipo di partizione LIST	452
Tipo di partizione DATE AUTO SPLIT	454
Utilizzo del partizionamento nativo	455
Lavorare con i LOB	456
Procedure ottimali e risoluzione dei problemi	457
Conversione del codice SQL di un'applicazione	459
Panoramica sulla conversione del codice SQL dell'applicazione	459
Conversione del codice SQL nelle tue applicazioni	460
Creazione di progetti generici di conversione di applicazioni	460
Gestione dei progetti di conversione delle applicazioni	465
Analizzare e convertire il codice SQL	466
Creazione e utilizzo del rapporto di valutazione	467
Modifica e salvataggio del codice SQL convertito	468
Conversione del codice SQL in applicazioni C#	468

Creazione di progetti di conversione di applicazioni C#	469
Conversione del codice SQL dell'applicazione C#	470
Salvare il codice dell'applicazione convertito	472
Gestione dei progetti di conversione di applicazioni C#	472
Creazione di un rapporto di valutazione della conversione delle applicazioni C#	473
Conversione del codice SQL in applicazioni C++	474
Creazione di progetti di conversione di applicazioni C++	475
Conversione del codice SQL dell'applicazione C++	476
Salvare il codice dell'applicazione convertito	478
Gestione dei progetti di conversione di applicazioni C++	479
Creazione di un rapporto di valutazione della conversione delle applicazioni C++	480
Conversione del codice SQL nelle applicazioni Java	481
Creazione di progetti di conversione di applicazioni Java	482
Conversione del codice SQL dell'applicazione Java	484
Salvare il codice dell'applicazione convertito	486
Gestione dei progetti di conversione di applicazioni Java	486
Creazione di un rapporto di valutazione della conversione delle applicazioni Java	487
Conversione del codice SQL nelle applicazioni Pro*C	489
Creazione di progetti di conversione di applicazioni Pro*C	489
Conversione del codice SQL dell'applicazione Pro*C	490
Modifica e salvataggio del codice dell'applicazione convertito	492
Gestione dei progetti di conversione delle applicazioni Pro*C	493
Creazione di un rapporto di valutazione della conversione delle applicazioni Pro*C	494
Utilizzo dei pacchetti di estensione	496
Autorizzazioni per l'utilizzo del pacchetto di estensione	497
Utilizzo dello schema del pacchetto di estensione	498
Librerie personalizzate per pacchetti di estensione	499
Applicazione del pacchetto di estensione	499
Utilizzo delle funzioni Lambda del AWS SCT pacchetto di estensione	502
Utilizzo di AWS Lambda funzioni per emulare la funzionalità del database	502
Applicazione del pacchetto di estensione per supportare le funzioni Lambda	502
Configurazione delle funzioni del pacchetto di estensione	504
Best practice	506
Configurazione della memoria aggiuntiva	506
Cartella di progetto predefinita	506
Incremento della velocità di migrazione dei dati dei dati	507

Aumento delle informazioni di registrazione	507
Risoluzione dei problemi	510
Impossibile caricare oggetti da un database di origine Oracle	510
Avvertenza	510
Documentazione di riferimento dell'interfaccia a riga di comando	512
Prerequisiti	512
Modalità interattiva	512
Esempi	514
Ottenere scenari CLI	514
Esempi	518
Modifica degli scenari CLI	518
Modalità script	520
Esempi	521
Materiale di riferimento	521
Note di rilascio	522
Note di rilascio: 675	522
Note di rilascio: 674	524
Note di rilascio: 673	532
Note di rilascio: 672	537
Note di rilascio: 671	545
Note di rilascio: 670	554
Note di rilascio: 669	559
Note di rilascio: 668	564
Note di rilascio — 667	571
Note di rilascio: 666	575
Note di rilascio: 665	580
Note di rilascio: 664	583
Note di rilascio: 663	587
Note di rilascio: 662	590
Note di rilascio: 661	595
Note di rilascio — 660	600
Note di rilascio: 659	604
Note di rilascio: 658	608
Note di rilascio: 657	613
Note di rilascio: 656	618
Note di rilascio — 655	621

Note di rilascio: 654	624
Note di rilascio: 653	628
Note di rilascio: 652	630
Note di rilascio: 651	632
Note di rilascio — 650	634
Note di rilascio — 649	636
Note di rilascio — 648	639
Note di rilascio — 647	641
Note di rilascio — 646	643
Note di rilascio — 645	645
Note di rilascio — 644	646
Note di rilascio — 642	648
Note di rilascio — 641	650
Note di rilascio: 640	651
Modifiche alla release 1.0.640 di Oracle	651
Modifiche alla release 1.0.640 di Microsoft SQL Server	657
Cambiamenti di MySQL versione 1.0.640	661
Modifiche a PostgreSQL nella release 1.0.640	662
Modifiche alla versione 1.0.640 Db2 LUW	665
Modifiche alla versione 1.0.640 di Teradata	666
Modifiche alla versione 1.0.640 per altri motori	667
Cronologia dei documenti	670
Aggiornamenti precedenti	687
.....	dcxcv

Che cosa è la AWS Schema Conversion Tool?

Puoi utilizzare AWS Schema Conversion Tool (AWS SCT) per convertire lo schema del database esistente da un motore di database a un altro. Puoi convertire lo schema OLTP relazionale o lo schema del data warehouse. Lo schema convertito è adatto per Amazon Relational Database Service (Amazon RDS) MySQL, MariaDB, Oracle, SQL Server, database PostgreSQL, cluster Amazon Aurora DB o cluster Amazon Redshift. Lo schema convertito può anche essere usato con un database su un'istanza Amazon EC2 o memorizzato sotto forma di dati su un bucket Amazon S3.

AWS SCT supporta diversi standard di settore, tra cui Federal Information Processing Standards (FIPS), per le connessioni a un bucket Amazon S3 o a un'altra risorsa. AWS SCT è inoltre conforme al Federal Risk and Authorization Management Program (FedRAMP). Per ulteriori informazioni su AWS e sulle strategie di conformità, consulta la pagina relativa ai [servizi AWS coperti dal programma di conformità](#).

AWS SCT supporta le seguenti conversioni OLTP.

Database di origine	Database di destinazione
IBM Db2 per z/OS (versione 12)	Edizione compatibile con Amazon Aurora MySQL (Aurora MySQL), edizione compatibile con Amazon Aurora PostgreSQL (Aurora PostgreSQL), MySQL, PostgreSQL Per ulteriori informazioni, consulta Utilizzo di IBM Db2 for z/OS come fonte .
IBM Db2 LUW (versioni 9.1, 9.5, 9.7, 10.5, 11.1 e 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL Per ulteriori informazioni, consulta Utilizzo di IBM Db2 LUW come origine .
Database SQL di Microsoft Azure	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL Per ulteriori informazioni, consulta Usare il database SQL di Azure come origine .

Database di origine	Database di destinazione
Microsoft SQL Server (versione 2008 R2, 2012, 2014, 2016, 2017, 2019 e 2022)	Aurora MySQL, Aurora PostgreSQL, Babelfish per Aurora PostgreSQL (solo per i rapporti di valutazione), MariaDB, Microsoft SQL Server, MySQL, PostgreSQL Per ulteriori informazioni, consulta Utilizzo di SQL Server come origine .
MySQL (versione 5.5 e successive)	Aurora PostgreSQL, MySQL, PostgreSQL Per ulteriori informazioni, consulta Utilizzo di MySQL come origine . È possibile migrare schemi e dati da MySQL a un cluster Aurora MySQL DB senza utilizzarli. AWS SCT Per ulteriori informazioni, consulta Migrazione dei dati su un cluster Amazon Aurora DB .
Oracle (versione 10.1 e successive)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL Per ulteriori informazioni, consulta Utilizzo del database Oracle come fonte .
PostgreSQL (versione 9.1 e successive)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL Per ulteriori informazioni, consulta Utilizzo di PostgreSQL come origine .
SAP ASE (versioni 12.5.4, 15.0.2, 15.5, 15.7 e 16.0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL Per ulteriori informazioni, consulta Utilizzo di SAP ASE (Sybase ASE) come fonte .

AWS SCT supporta le seguenti conversioni di data warehouse.

Data warehouse di origine	Data warehouse di destinazione
Amazon Redshift	Amazon Redshift Per ulteriori informazioni, consulta Utilizzo di Amazon Redshift come fonte .
Analisi di Azure Synapse	Amazon Redshift Per ulteriori informazioni, consulta Usare Azure Synapse Analytics come fonte .
BigQuery	Amazon Redshift Per ulteriori informazioni, consulta Utilizzo di BigQuery come origine .
Greenplum Database (versioni 4.3 e 6.21)	Amazon Redshift Per ulteriori informazioni, consulta Utilizzo del database Greenplum come fonte .
Microsoft SQL Server (versione 2008 e successive)	Amazon Redshift Per ulteriori informazioni, consulta Utilizzo di SQL Server Data Warehouse come origine .
Netezza (versione 7.0.3 e successive)	Amazon Redshift Per ulteriori informazioni, consulta Usare Netezza come fonte .
Oracle (versione 10.1 e successive)	Amazon Redshift Per ulteriori informazioni, consulta Utilizzo di Oracle Data Warehouse come origine .
Snowflake (versione 3)	Amazon Redshift

Data warehouse di origine	Data warehouse di destinazione
	Per ulteriori informazioni, consulta Usare Snowflake come fonte .
Teradata (versione 13 e successive)	Amazon Redshift Per ulteriori informazioni, consulta Utilizzo di Teradata come fonte .
Vertica (versione 7.2.2 e successive)	Amazon Redshift Per ulteriori informazioni, consulta Usare Vertica come fonte .

AWS SCT supporta le seguenti conversioni di database NoSQL di dati.

Database di origine	Database di destinazione
Apache Cassandra (versioni 2.1.x, 2.2.16 e 3.11.x)	Amazon DynamoDB Per ulteriori informazioni, consulta Usare Apache Cassandra come fonte .

AWS SCT supporta le conversioni dei seguenti processi di estrazione, trasformazione e caricamento (ETL). Per ulteriori informazioni, consulta [Conversione dei processi ETL](#).

Origine	Target
Script ETL di Informatica	Informatica
Pacchetti ETL di Microsoft SQL Server Integrati on Services (SSIS)	AWS Glue o AWS Glue Studio
Script di shell con comandi incorporati da Teradata Basic Teradata Query (BTEQ)	RSQL Amazon Redshift
Script ETL Teradata BTEQ	AWS Glue o Amazon Redshift RSQL

Origine	Target
Script di lavoro Teradata FastExport	RSQL Amazon Redshift
Script di lavoro Teradata FastLoad	RSQL Amazon Redshift
Script di lavoro Teradata MultiLoad	RSQL Amazon Redshift

AWS SCT supporta le seguenti migrazioni di framework di big data. Per ulteriori informazioni, consulta [Migrazione di framework per big data](#).

Origine	Target
Apache Hive (versione 0.13.0 e successive)	Hive su Amazon EMR
Apache HDFS	Amazon S3 o HDFS su Amazon EMR
Apache Oozie	AWS Step Functions

Panoramica della conversione dello schema

AWS SCT fornisce un'interfaccia utente basata su progetti per convertire automaticamente lo schema del database di origine in un formato compatibile con l'istanza Amazon RDS di destinazione. Se lo schema del database di origine non può essere convertito automaticamente, AWS SCT fornisce indicazioni su come creare uno schema equivalente nel database Amazon RDS di destinazione.

Per informazioni su come installare AWS SCT, consulta [Installazione, verifica e aggiornamento AWS SCT](#).

Per un'introduzione all'interfaccia utente di AWS SCT, consulta [Utilizzo dell'interfaccia AWS SCT utente](#).

Per informazioni sul processo di conversione, consulta [Conversione di schemi di database utilizzando AWS SCT](#).

Oltre a convertire lo schema del database esistente da un motore di database a un altro, AWS SCT dispone di alcune funzionalità aggiuntive che consentono di spostare dati e applicazioni nel AWS Cloud:

- Puoi utilizzare agenti di estrazione dei dati per estrarre i dati dal tuo data warehouse e prepararti alla migrazione su Amazon Redshift. Per gestire gli agenti di estrazione dei dati, puoi usare AWS SCT. Per ulteriori informazioni, consulta [Migrazione dei dati da un data warehouse locale ad Amazon Redshift](#).
- Puoi utilizzare AWS SCT per creare endpoint e attività di AWS DMS. Puoi eseguire e monitorare queste attività da AWS SCT. Per ulteriori informazioni, consulta [Uso di AWS SCT con AWS DMS](#).
- In alcuni casi, le funzionalità del database non possono essere convertite in funzionalità equivalenti di Amazon RDS o Amazon Redshift. La procedura guidata per il pacchetto di estensione AWS SCT può aiutarti a installare le funzioni AWS Lambda e le librerie Python per emulare le caratteristiche che non possono essere convertite. Per ulteriori informazioni, consulta [Utilizzo dei pacchetti di AWS SCT estensione](#).
- Puoi utilizzarlo AWS SCT per ottimizzare il tuo database Amazon Redshift esistente. AWS SCT consiglia chiavi di ordinamento e chiavi di distribuzione per ottimizzare il database. Per ulteriori informazioni, consulta [Ottimizzazione di Amazon Redshift utilizzando AWS SCT](#).
- Puoi utilizzarlo AWS SCT per copiare lo schema del database locale esistente su un'istanza database Amazon RDS che esegue lo stesso motore. Puoi utilizzare questa caratteristica per analizzare il potenziale risparmio sui costi del passaggio al cloud e della modifica del tipo di licenza.
- Puoi utilizzare AWS SCT per convertire SQL in codice C++, C#, Java o altro tipo di codice dell'applicazione. Puoi visualizzare, analizzare, modificare e salvare il codice SQL convertito. Per ulteriori informazioni, consulta [Conversione del codice SQL dell'applicazione utilizzando AWS SCT](#).
- È possibile utilizzarlo AWS SCT per migrare i processi di estrazione, trasformazione e caricamento (ETL). Per ulteriori informazioni, consulta [Conversione dei processi di estrazione, trasformazione e caricamento \(ETL\) con AWS Schema Conversion Tool](#).

Fornire feedback

Puoi fornire feedback su AWS SCT. Puoi segnalare un bug, inviare una richiesta di funzionalità o fornire informazioni generali.

Per fornire feedback su AWS SCT

1. Avvia il AWS Schema Conversion Tool.
2. Apri il menu Help (Aiuto) e quindi scegli Leave Feedback (Lascia un feedback). Viene visualizzata la finestra di dialogo Leave Feedback (Lascia feedback).

3. Per Area (Area), scegli Information (Informazioni), Bug report (Report sui bug) o Feature request (Richiesta di funzionalità).
4. Per Source database (Database di origine), scegli il tuo database di origine. Scegli Any (Qualsiasi) se il feedback non si riferisce a un database specifico.
5. Per Target database (Database di destinazione), scegli il tuo database di destinazione. Scegli Any (Qualsiasi) se il feedback non si riferisce a un database specifico.
6. Per Title (Titolo), digita un titolo per il tuo feedback.
7. Per Message (Messaggio), digita il tuo feedback.
8. Scegli Send (Invia) per inviare il tuo feedback.

Installazione, verifica e aggiornamento AWS SCT

AWS Schema Conversion Tool (AWS SCT) è un'applicazione autonoma che fornisce un'interfaccia utente basata su un progetto. AWS SCT è disponibile per Microsoft Windows, Fedora Linux e Ubuntu Linux. AWS SCT è supportato solo su sistemi operativi a 64 bit.

Per assicurarci di ottenere la versione corretta del file di AWS SCT distribuzione, forniamo i passaggi di verifica dopo aver scaricato il file compresso. Puoi verificare il file utilizzando i passaggi forniti.

Argomenti

- [Installazione AWS SCT](#)
- [Verifica del download del AWS SCT file](#)
- [Scaricamento dei driver di database richiesti](#)
- [Aggiornamento AWS SCT](#)
- [AWS SCT CLI](#)

Installazione AWS SCT

È possibile eseguire l'installazione AWS SCT sui seguenti sistemi operativi:

- Microsoft Windows 10
- Fedora Linux 36 e versioni successive
- Ubuntu Linux 18 e versioni successive

Per installare AWS SCT

1. Scaricate il file compresso che contiene il AWS SCT programma di installazione utilizzando il collegamento del sistema operativo in uso. Tutti i file compressi hanno un'estensione .zip. Quando estrai il file di AWS SCT installazione, sarà nel formato appropriato per il tuo sistema operativo.
 - [Microsoft Windows](#)
 - [Ubuntu Linux \(.deb\)](#)
 - [Fedora Linux \(.rpm\)](#)
2. Estrai il file di AWS SCT installazione per il tuo sistema operativo, mostrato di seguito.

Sistema operativo	Nome 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

- Eseguite il file di AWS SCT installazione estratto nel passaggio precedente. Utilizza le istruzioni per il tuo sistema operativo riportate di seguito.

Sistema operativo	Istruzioni di installazione
Fedora Linux	Esegui il comando seguente nella cartella in cui hai scaricato il file: <pre>sudo yum install aws-schema-conversion-tool-1.0. <i>build-number</i>.x86_64.rpm</pre>
Microsoft Windows	Fai doppio clic sul file per eseguire il programma di installazione.
Ubuntu Linux	Esegui il comando seguente nella cartella in cui hai scaricato il file: <pre>sudo dpkg -i aws-schema-conversion-tool-1.0. <i>build-number</i>.deb</pre>

- Scaricate i driver Java Database Connectivity (JDBC) per i motori di database di origine e di destinazione. Per istruzioni e collegamenti per il download, consulta [Scaricamento dei driver di database richiesti](#).

Ora hai completato la configurazione dell' AWS SCT applicazione. Fate doppio clic sull'icona dell'applicazione per AWS SCT eseguirla.

Installazione delle versioni precedenti di AWS SCT

È possibile scaricare e installare versioni precedenti AWS SCT a partire dalla 1.0.625. Per scaricare una versione precedente, fornisci le informazioni sulla versione e sul sistema operativo utilizzando il seguente formato.

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

Ad esempio, per scaricare la AWS SCT versione 668, procedi come segue:

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

AWS supporta le tre ultime versioni dell'applicazione. AWS SCT Si consiglia di utilizzare la versione più recente di AWS SCT.

Verifica del download del AWS SCT file

Esistono diversi modi per verificare il file di distribuzione di AWS SCT. Il metodo più semplice consiste nel confrontare il checksum del file con il checksum pubblicato da AWS. Come livello di sicurezza aggiuntivo, puoi utilizzare le procedure di seguito per verificare il file di distribuzione in base al sistema operativo in cui hai installato il file.

Questa sezione contiene gli argomenti seguenti.

Argomenti

- [Verifica del checksum del file AWS SCT](#)
- [Verifica dei file AWS SCT RPM su Fedora](#)
- [Verifica dei file AWS SCT DEB su Ubuntu](#)
- [Verifica del file AWS SCT MSI su Microsoft Windows](#)

Verifica del checksum del file AWS SCT

Per rilevare eventuali errori che potrebbero essere stati introdotti durante il download o la memorizzazione del file AWS SCT compresso, è possibile confrontare il checksum del file con un valore fornito da AWS. AWS utilizza l'algoritmo SHA256 per il checksum.

Per verificare il file di AWS SCT distribuzione utilizzando un checksum

1. Scaricate il file di AWS SCT distribuzione utilizzando i collegamenti nella sezione Installazione. Per ulteriori informazioni, consulta [Installazione AWS SCT](#).
2. Scarica la versione più recente del file di checksum, denominata [sha256Check.txt](#). Questo file include i checksum per la versione più recente AWS SCT . Ad esempio, il file può apparire come segue:

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

3. Esegui il comando di convalida SHA256 per il tuo sistema operativo nella directory che contiene i file di distribuzione. Ad esempio, esegui il comando seguente in Linux.

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

4. Confronta i risultati del comando con il valore illustrato nel file sha256Check.txt. Se i checksum corrispondono, è sicuro eseguire il file di distribuzione. Se i checksum non corrispondono, non eseguire il file di distribuzione e contatta [AWS Support](#).

Verifica dei file AWS SCT RPM su Fedora

AWS fornisce un altro livello di convalida oltre al checksum del file di distribuzione. Tutti i file RPM nel file di distribuzione sono firmati da una AWS chiave privata. Si può visualizzare La chiave pubblica GPG in [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key).

Per verificare i file AWS SCT RPM su Fedora

1. Scarica il file AWS SCT di distribuzione usando i link nella sezione Installazione.
2. Verifica il checksum del file di AWS SCT distribuzione.
3. Estrai il contenuto del file di distribuzione. Individua il file RPM che desideri verificare.
4. Scarica la chiave pubblica GPG da [amazon.com.public.gpg-key](https://amazon.com/public.gpg-key)
5. Importa la chiave pubblica nel database RPM (assicurati di disporre delle autorizzazioni appropriate) utilizzando il comando seguente:


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

6. Controlla l'esito positivo dell'importazione eseguendo il seguente comando:

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

7. Controlla la firma RPM eseguendo il seguente comando:

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

Verifica dei file AWS SCT DEB su Ubuntu

AWS fornisce un altro livello di convalida oltre al checksum del file di distribuzione. Tutti i file DEB nel file di distribuzione sono firmati da una firma disconnessa di GPG.

Per verificare i file AWS SCT DEB su Ubuntu

1. Scarica il file AWS SCT di distribuzione utilizzando i collegamenti nella sezione Installazione.
2. Verifica del checksum del file di AWS SCT distribuzione.
3. Estrai il contenuto del file di distribuzione. Individua il file DEB che desideri verificare.
4. [Scarica la firma separata da -1.0.latest.deb.asc. aws-schema-conversion-tool](#)
5. Scarica la chiave pubblica GPG da [amazon.com.public.gpg-key](#).
6. Importa la chiave pubblica GPG eseguendo il seguente comando:

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

7. Verifica la firma eseguendo il seguente comando:

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

Verifica del file AWS SCT MSI su Microsoft Windows

AWS fornisce un altro livello di convalida oltre al checksum del file di distribuzione. Il file MSI ha una firma digitale che puoi verificare per assicurarti che sia stata firmata. AWS

Per verificare il file AWS SCT MSI su Windows

1. Scaricate il file di AWS SCT distribuzione utilizzando i collegamenti nella sezione Installazione.
2. Verifica del checksum del file di AWS SCT distribuzione.
3. Estrai il contenuto del file di distribuzione. Individua il file MSI che desideri verificare.
4. In Windows Explorer fai clic con il pulsante destro sul file MSI e seleziona Proprietà.
5. Scegli la scheda Firme digitali.
6. Verifica che la firma digitale provenga da Amazon Services LLC.

Scaricamento dei driver di database richiesti

AWS SCT Per un corretto funzionamento, scaricate i driver JDBC per i motori di database di origine e di destinazione. Se utilizzi una piattaforma di database di destinazione virtuale, non è necessario scaricare il driver JDBC per il motore di database di destinazione. Per ulteriori informazioni, consulta [Utilizzo di obiettivi virtuali](#).

Dopo aver scaricato i driver, puoi fornire il percorso dei file driver. Per ulteriori informazioni, consulta [Memorizzazione dei percorsi dei driver nelle impostazioni globali](#).

Puoi scaricare i driver del database dai percorsi seguenti.

Important

Scarica la versione più recente del driver disponibile. La tabella seguente include la versione più bassa del driver di database supportata da AWS SCT.

Motore del database	Driver	Posizione di download
Amazon Aurora edizione compatibile con MySQL	mysql-connector-java-5.1.6.jar	https://www.mysql.com/products/connector/

Motore del database	Driver	Posizione di download
Amazon Aurora PostgreSQL-Compatible Edition	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 Serverless	redshift-jdbc42-2.1.0.9.jar	https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
Apache Hive	hive-jdbc-2.3.4-standalone.jar	https://repo1.maven.org/maven2/org/apache/hive/hive-jdbc/2.3.4/hive-jdbc-2.3.4-standalone.jar
Database SQL di Azure	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/release-notes-for-the-jdbc-driver?sql-server-vervista=15#72
Analisi di Azure Synapse	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/sql/connect/jdbc/-jdbc-driver?release-notes-for-the-sql-server-vervista=15#72
Database Greenplum	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
IBM Db2 per z/OS	db2jcc-db2jcc4.jar	https://www.ibm.com/support/pages/db2-jdbc-driver-versions-and-downloads-db2-zos
IBM Db2 LUW	db2jcc-db2jcc4.jar	https://www.ibm.com/support/pages/node/382667

Motore del database	Driver	Posizione di download
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&visualstudio=15&sql-server-ver
MySQL	mysql-connector-java-8.0.15.jar	https://dev.mysql.com/downloads/connector/j/
Netezza	nzjdbc.jar Utilizza il software per il tool client. Scarica la versione del driver 7.2.1, che è retrocompatibile con la versione 7.2.0 del data warehouse.	http://www.ibm.com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.datacon.doc/c_datacon_plg_overview.html
Oracle	ojdbc8.jar Sono supportate le versioni 8 e successive dei driver.	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/sq1a170/en/html/3bd588c76c5f10149824867aa281fadd.html

Motore del database	Driver	Posizione di download
Snowflake	snowflake-jdbc-3.9.2.jar Per ulteriori informazioni, vedere Download e integrazioni del driver JDBC .	https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9.2.jar
Teradata	terajdbc4.jar tdgssconfig.jar Per la versione 16.20.00.11 e successive del driver Teradata JDBC, non è necessario il file. tdgssconfig.jar	https://downloads.teradata.com/download/connectivity/jdbc-driver
Vertica	vertica-jdbc-9.1.1-0.jar Sono supportate le versioni dei driver 7.2.0 e successive.	https://www.vertica.com/client_drivers/9.1.x/9.1.1-0/vertica-jdbc-9.1.1-0.jar

Installazione dei driver JDBC su Linux

È possibile utilizzare i seguenti passaggi per installare i driver JDBC sul sistema Linux da utilizzare con AWS SCT

Installazione dei driver JDBC sul sistema Linux

1. Crea una directory in cui archiviare i driver JDBC.

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

2. Installa il driver JDBC per il motore di database utilizzando i comandi illustrati di seguito.

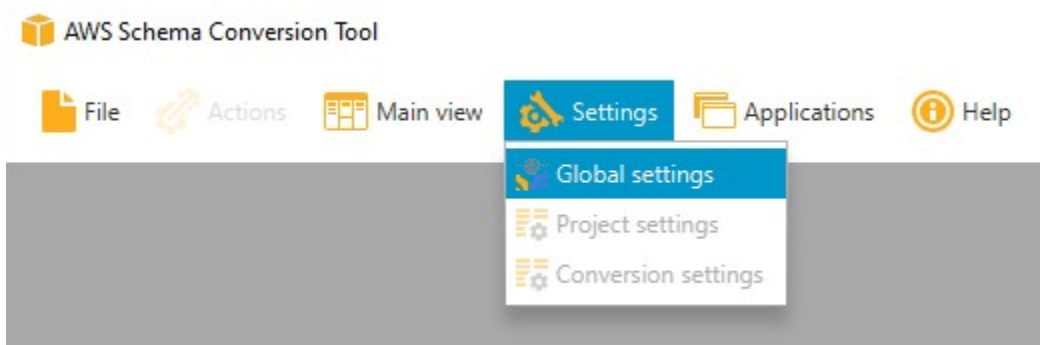
Motore del database	Comandi di installazione
Amazon Aurora (compatibile con MySQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X .X.X.tar.gz</pre>
Amazon Aurora (compatibile con PostgreSQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>
Microsoft SQL Server	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/sqljdbc_X.X.X_enu.tar.gz</pre>
MySQL	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X .X.X.tar.gz</pre>
Oracle	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo mkdir oracle-jdbc PROMPT> cd oracle-jdbc PROMPT> sudo cp -a /tmp/ojdbc8.jar .</pre>
PostgreSQL	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>

Memorizzazione dei percorsi dei driver nelle impostazioni globali

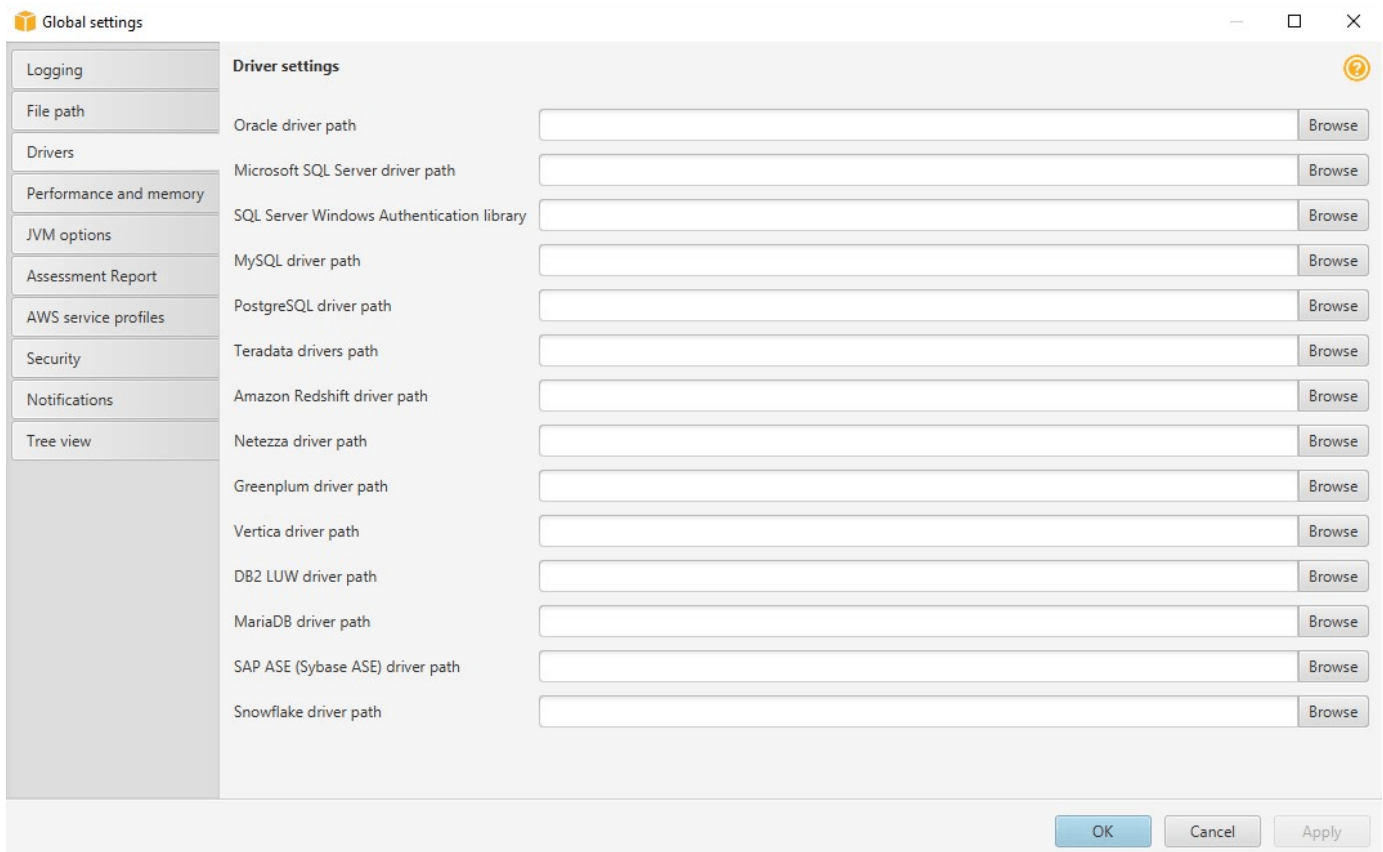
Dopo aver scaricato e installato i driver JDBC richiesti, puoi impostare la posizione dei driver a livello globale nelle AWS SCT impostazioni. Se non imposti la posizione del driver a livello globale, l'applicazione ti richiede il percorso dei driver quando ti connetti a un database.

Aggiornamento delle posizioni del file driver

1. In AWS SCT, scegli Impostazioni, quindi scegli Impostazioni globali.



2. In Global settings (Impostazioni globali), scegli Driver. Aggiungi il percorso del file al driver JDBC per il motore di database di origine e il motore di database di istanze Amazon RDS DB di destinazione.



3. Quando hai finito di aggiungere i percorsi di driver, seleziona OK.

Aggiornamento AWS SCT

AWS aggiornamenti periodici AWS SCT con nuove caratteristiche e funzionalità. Se state eseguendo l'aggiornamento da una versione precedente, create un nuovo AWS SCT progetto e riconvertite tutti gli oggetti di database che state utilizzando.

Puoi verificare se esistono aggiornamenti per AWS SCT.

Per verificare la presenza di aggiornamenti a AWS SCT

1. Una volta entrato AWS SCT, scegli Aiuto, quindi scegli Verifica la disponibilità di aggiornamenti.
2. Nella finestra di dialogo Check for Updates (Controlla aggiornamenti) seleziona Novità. Se il link non viene visualizzato, hai la versione più recente.

AWS SCT CLI

È possibile scaricare la AWS SCT CLI per l'utilizzo da riga di comando. Per scaricare il JAR, usa il seguente link:

[AWSSchemaConversionToolBatch.jar](#)

Utilizzo dell'interfacciaAWS SCT utente

Utilizza i seguenti argomenti per informazioni su come lavorare con l'interfacciaAWS SCT utente. Per informazioni sull'installazioneAWS SCT, vedere [Installazione, verifica e aggiornamento AWS SCT](#).

Argomenti

- [La finestraAWS SCT del progetto](#)
- [Avvio di AWS SCT](#)
- [Creare unAWS SCT progetto](#)
- [Utilizzo di una nuova procedura guidata di progetto inAWS SCT](#)
- [Salvare e aprire unAWS SCT progetto](#)
- [Aggiungere server di database a unAWS SCT progetto](#)
- [EsecuzioneAWS SCT in modalità offline](#)
- [Utilizzo di filtriAWS SCT ad albero](#)
- [Nascondere gli schemi nella vistaAWS SCT ad albero](#)
- [Creazione e revisione del rapporto di valutazione della migrazione del database](#)
- [Conversione dello schema](#)
- [Applicazione dello schema convertito all'istanza database di destinazione](#)
- [Archiviazione dei profiliAWS di servizio inAWS SCT](#)
- [Utilizzo di AWS Secrets Manager](#)
- [Memorizzare le password del database](#)
- [Utilizzo della vista UNION ALL per progetti con tabelle partizionate](#)
- [Scelte rapide da tastiera perAWS SCT](#)

La finestraAWS SCT del progetto

L'illustrazione che segue è ciò che si vedeAWS SCT quando si crea un progetto di migrazione dello schema e quindi si converte uno schema.

1. Nel pannello a sinistra, è presentato lo schema dal database di origine in una visualizzazione ad albero. Lo schema del database è a "caricamento differito." In altre parole, quando selezioni una

- voce della visualizzazione ad albero, AWS SCT individua e visualizza lo schema corrente dal database di origine.
2. Nel pannello centrale superiore, compaiono le attività per elementi dello schema dal motore di database di origine che non è stato possibile convertire automaticamente per il motore di database di destinazione.
 3. Nel pannello a destra, è presentato lo schema dall'istanza database di destinazione in una visualizzazione ad albero. Lo schema del database è a "caricamento differito." Ciò significa che quando selezioni una voce della visualizzazione ad albero, AWS SCT individua e visualizza lo schema corrente dal database di destinazione.

The screenshot shows the AWS Schema Conversion Tool interface. The interface is divided into several panes:

- 1** (Left): A tree view showing the source Microsoft SQL Server schema structure, including databases like 'Adventureworks' and 'SampleDatabase', and tables like 'Customers' and 'Address'.
- 2** (Center): A list of conversion issues. The highlighted issue is: "Issue: 497: MySQL doesn't support an Identity column outside a primary key". Below the list, there are two table comparison panes for the 'Address' table, showing properties and SQL definitions for both the source and target databases.
- 3** (Right): A tree view showing the target Amazon RDS for MySQL schema structure, including schemas like 'information_schema' and 'mysql', and tables like 'Address'.
- 4** (Bottom Left): A circular callout pointing to the 'Address' table in the source table comparison pane.
- 5** (Bottom Right): A circular callout pointing to the 'Address' table in the target table comparison pane.

At the bottom of the screenshot, the following memory usage information is displayed: Used memory: 120.67 MB, Free memory: 39.83 MB, Total memory: 160.5 MB, Maximum memory: 884 MB.

4. Nel pannello in basso a sinistra, quando si sceglie un elemento dello schema, vengono visualizzate le proprietà. Questi descrivono l'elemento dello schema di origine e il comando SQL per creare quell'elemento nel database di origine.
5. Nel pannello in basso a destra, quando si sceglie un elemento dello schema, vengono visualizzate le proprietà. Questi descrivono l'elemento dello schema di destinazione e il comando SQL per creare quell'elemento nel database di destinazione. Puoi modificare tale comando SQL e salvare il comando aggiornato con il progetto.

Avvio di AWS SCT

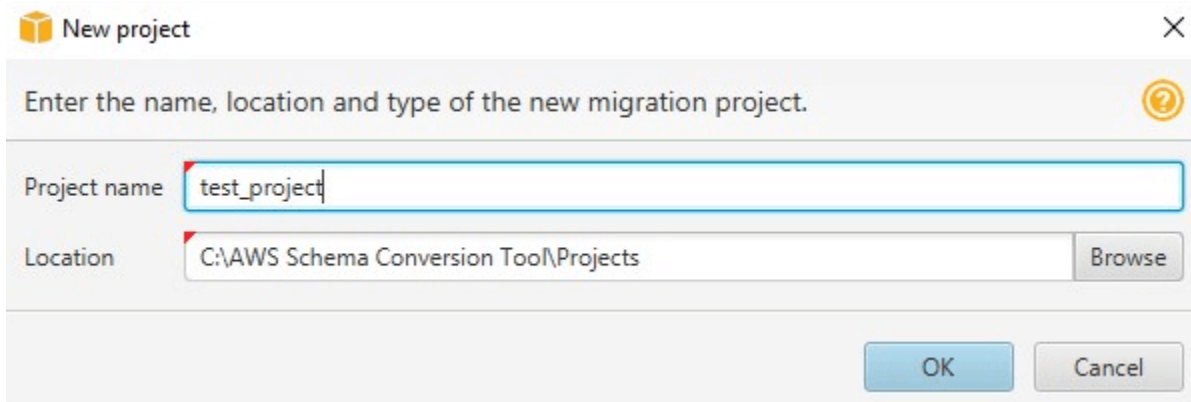
Per avviare ilAWS Schema Conversion Tool, fare doppio clic sull'icona dell'applicazione.

Creare unAWS SCT progetto

Per creare unAWS Schema Conversion Tool progetto, utilizza la procedura seguente.

Creazione del tuo progetto

1. Avvia il AWS Schema Conversion Tool.
2. Nel menu File, scegli Nuovo progetto. Viene visualizzata la finestra di dialogo Nuovo progetto.



3. Immetti un nome per il progetto, che è memorizzato localmente nel computer.
4. Immetti l'ubicazione del file del progetto locale.
5. Scegli OK per creare il progetto AWS SCT.
6. Scegli Aggiungi fonte per aggiungere un nuovo database di origine al tuoAWS SCT progetto. Puoi aggiungere più database di origine al tuoAWS SCT progetto.
7. Scegli Aggiungi obiettivo per aggiungere una nuova piattaforma di destinazione nel tuoAWS SCT progetto. Puoi aggiungere più piattaforme target al tuoAWS SCT progetto.

8. Scegli lo schema del database di origine nel pannello di sinistra.
9. Nel pannello di destra, specificare la piattaforma del database di destinazione per lo schema di origine selezionato.
10. Scegli Crea mappatura. Questo pulsante diventa attivo dopo aver scelto lo schema del database di origine e la piattaforma del database di destinazione. Per ulteriori informazioni, consulta [Creazione di regole di mappatura](#).

Ora, il tuo AWS SCT progetto è impostato. È possibile salvare il progetto, creare un rapporto di valutazione della migrazione del database e convertire gli schemi del database di origine.

Utilizzo di una nuova procedura guidata di progetto in AWS SCT

È possibile creare un nuovo progetto di migrazione del database utilizzando la nuova procedura guidata del progetto. Questa procedura guidata ti aiuta a determinare l'obiettivo di migrazione e a connetterti ai tuoi database. Stima la complessità di una migrazione per tutte le destinazioni target supportate. Dopo aver eseguito la procedura guidata, AWS SCT produce un rapporto di riepilogo per la migrazione del database verso diverse destinazioni di destinazione. Puoi utilizzare questo rapporto per confrontare le possibili destinazioni di destinazione e scegliere il percorso di migrazione ottimale.

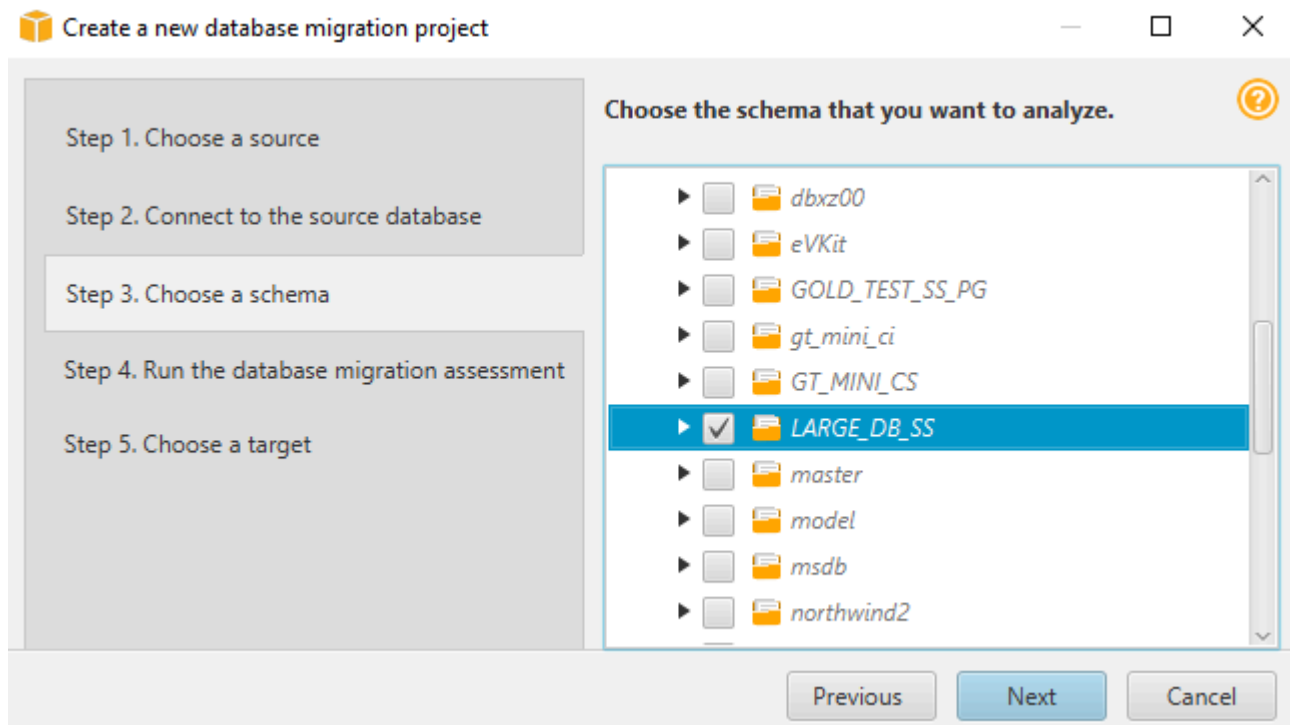
Per eseguire la procedura guidata del nuovo progetto

1. Scegli il tuo database di origine.
 - a. Avvia il AWS Schema Conversion Tool.
 - b. Nel menu File, scegli Nuova procedura guidata per il progetto. Viene visualizzata la finestra di dialogo Crea un nuovo progetto di migrazione del database.
 - c. Per inserire le informazioni di connessione al database di origine, utilizza le istruzioni seguenti:

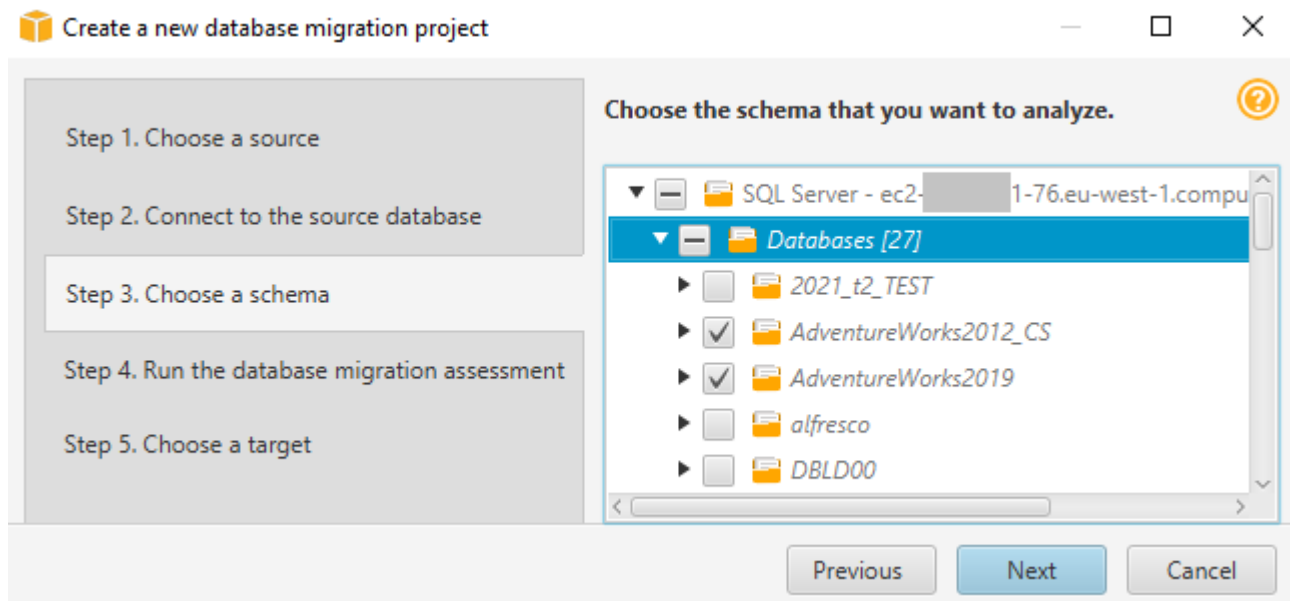
Parametro	Operazione
Project name (Nome progetto)	Immetti un nome per il progetto, che è memorizzato localmente nel computer.
Location (Ubicazione)	Immetti l'ubicazione del file del progetto locale.

Parametro	Operazione
Source type (Tipo di origine)	<p>Scegli una delle seguenti opzioni: database SQL, database NoSQL o ETL.</p> <p>Se desideri visualizzare il rapporto di riepilogo che include tutte le destinazioni di migrazione, scegli il database SQL.</p>
Motore di origine	Scegli il motore di database di origine.
Strategia di migrazione	<p>Seleziona una delle seguenti opzioni:</p> <ul style="list-style-type: none"> • Voglio cambiare motore e ottimizzare per il cloud: questa opzione converte il database di origine in un nuovo motore di database. • Voglio mantenere lo stesso motore ma ottimizzarlo per il cloud: questa opzione mantiene il motore del database così com'è e sposta il database dall'ambiente locale al cloud. • Vorrei vedere un rapporto combinato per il passaggio al motore di database e l'ottimizzazione per il cloud: questa opzione confronta la complessità della migrazione di tutte le opzioni di migrazione disponibili. <p>Se desideri visualizzare il rapporto di valutazione aggregato che include tutte le destinazioni di migrazione, scegli l'ultima opzione.</p>

- d. Seleziona Successivo. Viene visualizzata la pagina Connect al database di origine.
2. Connettiti al tuo database di origine;
 - a. Fornire le informazioni di connessione per il database di origine. I parametri di connessione dipendono dal motore del database di origine. Assicurati che l'utente utilizzato per l'analisi del database di origine disponga delle autorizzazioni applicabili. Per ulteriori informazioni, consulta [Origini per AWS SCT](#).
 - b. Seleziona Next (Successivo). Viene visualizzata la pagina Scegli uno schema.
 3. Scegli lo schema del tuo database.
 - a. Seleziona la casella di controllo del nome degli schemi che desideri valutare, quindi scegli lo schema stesso. Il nome dello schema è evidenziato in blu quando selezionato e il pulsante Avanti è disponibile.



- b. Se desideri valutare diversi schemi di database, seleziona le caselle di controllo per tutti gli schemi e quindi scegli il nodo principale. Per una valutazione corretta, è necessario scegliere il nodo principale. Ad esempio, per un database SQL Server di origine, scegliere il nodo Databases. Il nome del nodo principale è evidenziato in blu e il pulsante Avanti è disponibile.



- c. Scegli Avanti. AWS SCT analizza gli schemi del database di origine e crea un rapporto di valutazione della migrazione del database. Il numero di oggetti del database negli schemi del

database di origine influisce sul tempo necessario per eseguire la valutazione. Al termine, viene visualizzata la pagina di valutazione. Esegui la migrazione del database.

4. Esegui la valutazione della migrazione del database.
 - a. È possibile esaminare e confrontare i report di valutazione per diversi obiettivi di migrazione o salvare una copia locale dei file dei report di valutazione per un'ulteriore analisi.
 - b. Salva una copia locale del rapporto di valutazione della migrazione del database. Scegli Salva, quindi inserisci il percorso della cartella per salvare i file e scegli Salva. AWS SCT salva i file del rapporto di valutazione nella cartella specificata.
 - c. Seleziona Successivo. Viene visualizzata la pagina Scegli un obiettivo.
5. Scegli il tuo database di destinazione.
 - a. Per Target engine, scegli il motore di database di destinazione che decidi di utilizzare in base al rapporto di valutazione.
 - b. Fornisci le informazioni di connessione per il database di destinazione. I parametri di connessione visualizzati dipendono dal motore di database di destinazione selezionato. Assicurati che l'utente specificato per il database di destinazione disponga delle autorizzazioni richieste. Per ulteriori informazioni sulle autorizzazioni richieste, consulta le sezioni che descrivono le autorizzazioni per i database di destinazione in [Origini per AWS SCT](#) e [Autorizzazioni per Amazon Redshift come obiettivo](#).
 - c. Scegli Fine. AWS SCT crea il progetto e aggiunge le regole di mappatura. Per ulteriori informazioni, consulta [Creazione di regole di mappatura](#).

Ora puoi usare ilAWS SCT progetto per convertire gli oggetti del database di origine.

Salvare e aprire unAWS SCT progetto

Per salvare unAWS Schema Conversion Tool progetto, utilizza la procedura seguente.

Per salvare il progetto

1. Avvia il AWS Schema Conversion Tool.
2. Nel menu File, scegli Salva progetto.

AWS SCTsalva il progetto nella cartella specificata al momento della creazione del progetto.

Per aprire unAWS Schema Conversion Tool progetto esistente, utilizza la procedura seguente.

Per aprire il progetto

1. Nel menu File, scegli Apri progetto. Viene visualizzata la finestra di dialogo Apri.
2. Scegliete la cartella del progetto e quindi il file Windows Script Component (*.sct).
3. AWS SCT apre il progetto ma non si connette automaticamente ai database di origine e di destinazione. Scegli Connect al server nella parte superiore degli alberi dello schema del database per connetterti ai database di origine e di destinazione.

Se si apre un progetto salvato nellaAWS SCT versione 1.0.655 o precedente, creaAWS SCT automaticamente regole di mappatura per tutti gli schemi del database di origine sulla piattaforma del database di destinazione. Per aggiungere altre piattaforme di database di destinazione, elimina le regole di mappatura esistenti e quindi crea nuove regole di mappatura. Per ulteriori informazioni sulla creazione di regole di mappatura, vedere[Creazione di regole di mappatura](#).

Aggiungere server di database a unAWS SCT progetto

È possibile aggiungere più server di database di origine e destinazione a unAWS Schema Conversion Tool progetto.

Per aggiungere un server al progetto

1. Avvia il AWS Schema Conversion Tool.
2. Crea un nuovo progetto o apri un progetto esistente.
3. Scegli Aggiungi fonte dal menu per aggiungere un nuovo database di origine.
4. Scegli una piattaforma di database e specifica le credenziali di connessione al database. Per ulteriori informazioni sulla connessione a un database di origine, consulta[Origini per AWS SCT](#).

Procedi come segue per stabilire una connessione a un database.

Connessione al tuo database

1. Apri il menu di scelta rapida (tasto destro del mouse) per un server di database e quindi scegli Stabilire connessione.

Puoi anche scegliere Connect al server nella parte superiore dell'albero dello schema del database.

2. Inserisci la password per connetterti al server del database di origine.
3. Scegli Verifica connessione per verificare che sia inAWS SCT grado di connettersi al database di origine.
4. Scegli Connect per connetterti al tuo database di origine.

Per rimuovere un server di database dalAWS SCT progetto, utilizza la procedura seguente.

Per rimuovere un server di database

1. Scegli il server di database da rimuovere.
2. Apri il menu di scelta rapida (tasto destro del mouse) e quindi scegli Rimuovi dal progetto.

AWS SCTrimuove il server di database selezionato, tutte le regole di mappatura, i risultati della conversione e altri metadati relativi a questo server.

EsecuzioneAWS SCT in modalità offline

Puoi eseguireAWS Schema Conversion Tool in modalità offline. Di seguito, viene illustrato come lavorare con unAWS SCT progetto esistente quando si è disconnessi dal database di origine.

AWS SCTnon richiede una connessione al database di origine per eseguire le seguenti operazioni:

- Aggiungi regole di mappatura.
- Crea report di valutazione della migrazione del database.
- Converti schemi e codice del database.
- Modifica il codice sorgente e quello convertito.
- Salva il codice sorgente e quello convertito come script SQL in un file di testo.

Prima di utilizzarloAWS SCT in modalità offline, connettiti al database di origine, carica i metadati e salva il progetto. Apri questo progetto o disconnettiti dal server del database di origine per utilizzarloAWS SCT in modalità offline.

Per eseguire AWS SCT in modalità offline

1. Iniziare il progetto AWS Schema Conversion Tool e creare un nuovo progetto. Per ulteriori informazioni, consulta [Creare un AWS SCT progetto](#).
2. Aggiungi un server del database di origine e connettiti al tuo database di origine. Per ulteriori informazioni, consulta [Aggiungere server di database a un AWS SCT progetto](#).
3. Aggiungi un server di database di destinazione o utilizza una piattaforma di database di destinazione virtuale. Per ulteriori informazioni, consulta [Utilizzo di obiettivi virtuali](#).
4. Crea una regola di mappatura per definire la piattaforma di database di destinazione per il tuo database di origine. Per ulteriori informazioni, consulta [Creazione di regole di mappatura in AWS SCT](#).
5. Scegliete Visualizza, quindi scegliete Vista principale.
6. Nel pannello di sinistra che mostra gli oggetti del database di origine, scegli gli schemi del database di origine. Apri il menu di scelta rapida (tasto destro del mouse) per l'oggetto e quindi scegli Carica schema. Questa operazione carica tutti i metadati dello schema di origine nel AWS SCT progetto.

Le operazioni Crea report e Converti schema caricano anche tutti i metadati dello schema di origine nel AWS SCT progetto. Se hai eseguito una di queste operazioni dal menu contestuale, salta l'operazione Carica schema.

7. Nel menu File, scegliete Salva progetto per salvare i metadati del database di origine nel progetto.
8. Scegli Disconnetti dal server per disconnetterti dal database di origine. Ora puoi usarlo AWS SCT in modalità offline.

Utilizzo di filtri AWS SCT ad albero

Per eseguire la migrazione di dati da un'origine a una destinazione, AWS SCT carica tutti i metadati dai database di origine e di destinazione in una struttura ad albero. Questa struttura compare in AWS SCT come visualizzazione ad albero nella finestra principale del progetto.

Alcuni database possono presentare un gran quantità di oggetti nella struttura ad albero. Puoi utilizzare i filtri di struttura in AWS SCT per cercare oggetti nelle strutture ad albero di origine e di destinazione. Quando utilizzi un filtro di struttura, non apporti modifiche agli oggetti convertiti durante la conversione del database. Il filtro modifica solo ciò che vedi nell'albero.

I filtri di struttura funzionano con gli oggetti precaricati da AWS SCT. In altre parole, AWS SCT non carica gli oggetti provenienti dal database durante le ricerche. Di conseguenza, la struttura ad albero in genere contiene un minor numero di oggetti di quelli presenti nel database.

Per i filtri di struttura, tieni presente quanto segue:

- Il filtro predefinito è ANY, il che significa che il filtro utilizza una ricerca per nome per individuare gli oggetti.
- Selezionando uno o più tipi di oggetto, visualizzerai nell'albero solo tali tipi di oggetto.
- Puoi utilizzare la maschera di filtro per visualizzare diversi tipi di simboli, tra cui Unicode, spazi e caratteri speciali. Il carattere “%” è il carattere jolly per qualsiasi simbolo.
- Dopo aver applicato un filtro, il conteggio mostra solo il numero di oggetti filtrati.

Creazione di un filtro di struttura

1. Apri un progetto AWS SCT esistente.
2. Connect al database a cui desideri applicare il filtro ad albero.
3. Seleziona l'icona Filtro.



L'icona di annullamento filtro è disabilitata in quanto nessun filtro è attualmente applicato.

4. Immettete le seguenti informazioni nella finestra di dialogo Filtro. Le opzioni nella finestra di dialogo sono differenti per ciascun motore di database.

AWS SCTopzione filtro	Operazione
Livello	Scegli Categorie per filtrare gli oggetti per categorie. Scegliete Stati per filtrare gli oggetti in base allo stato.
Type (Tipo)	Per Categorie in Level, scegli le categorie di oggetti filtrati. Scegli Qualsiasi caricato per visualizzare gli oggetti di tutte le categorie. In Statuses in Level, scegli lo stato degli oggetti filtrati. Puoi scegliere una delle seguenti opzioni:

AWS SCTopzione filtro	Operazione
	<ul style="list-style-type: none">• Convertito per visualizzare tutti gli oggetti convertiti• Dispone di azioni per visualizzare tutti gli oggetti che presentano problemi di conversione• Crittografato per visualizzare tutti gli oggetti crittografati
Condition	<p>Per Categorie di livello, scegli la condizione di filtro tra Mi piace e Non mi piace.</p> <p>Per gli stati in Level, l'opzione della condizione di filtro non è disponibile.</p>
Value (Valore)	<p>Per Categorie in Level, inserisci il valore per filtrare l'albero in base a questo valore.</p> <p>Usa la percentuale (%) come jolly per visualizzare tutti gli oggetti.</p> <p>Per gli stati in Level, scegli il valore tra Vero e Falso.</p>
e/o	Scegli iAND nostri operatoriOR logici per applicare più clausole di filtro.

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

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

5. Scegli Aggiungi nuova clausola per aggiungere una clausola di filtro aggiuntiva. AWS SCT può applicare più clausole di filtro utilizzando i nostri operatori OR logici.
6. Seleziona Apply (Applica). Dopo aver selezionato Apply (Applica), l'icona di annullamento filtro (accanto all'icona Filtro) viene abilitata. Utilizza questa icona per rimuovere i filtri applicati.
7. Seleziona Close (Chiudi) per chiudere la finestra di dialogo.

Quando applichi un filtro allo schema che compare nell'albero, non apporti modifiche agli oggetti convertiti durante la conversione dello schema. Il filtro modifica solo ciò che visualizzi nell'albero.

Importazione di un elenco di file per il filtro ad albero

È possibile importare un file con valori delimitati da virgole (CSV) con separatori di punto e virgola o un file JSON contenente nomi o valori che si desidera vengano utilizzati dal filtro ad albero. Apri un AWS SCT progetto esistente, connettiti al database a cui applicare il filtro ad albero, quindi scegli l'icona del filtro.

Per scaricare un esempio del file, scegli Scarica modello. Inserisci il nome del file e scegli Salva.

Per scaricare le impostazioni del filtro esistenti, scegli Esporta. Inserisci il nome del file e scegli Salva.

Per importare un elenco di file per il filtro ad albero, scegli Importa. Seleziona un file da importare, quindi selezionare Open (Apri). Seleziona Apply (Applica), quindi seleziona Close (Chiudi).

I file CSV utilizzano il punto e virgola come separatore e hanno il seguente formato:

- `object_type` è il tipo di oggetto che intendi trovare.
- `database_name` è il nome del database in cui esiste l'oggetto.
- `schema_name` è il nome dello schema in cui esiste l'oggetto.
- `object_name` è il nome oggetto.
- `import_type` specifica `include` o `exclude` questo elemento dal filtro.

Usa i file JSON per descrivere casi di filtraggio complessi, come le regole annidate. I file JSON hanno il seguente formato:

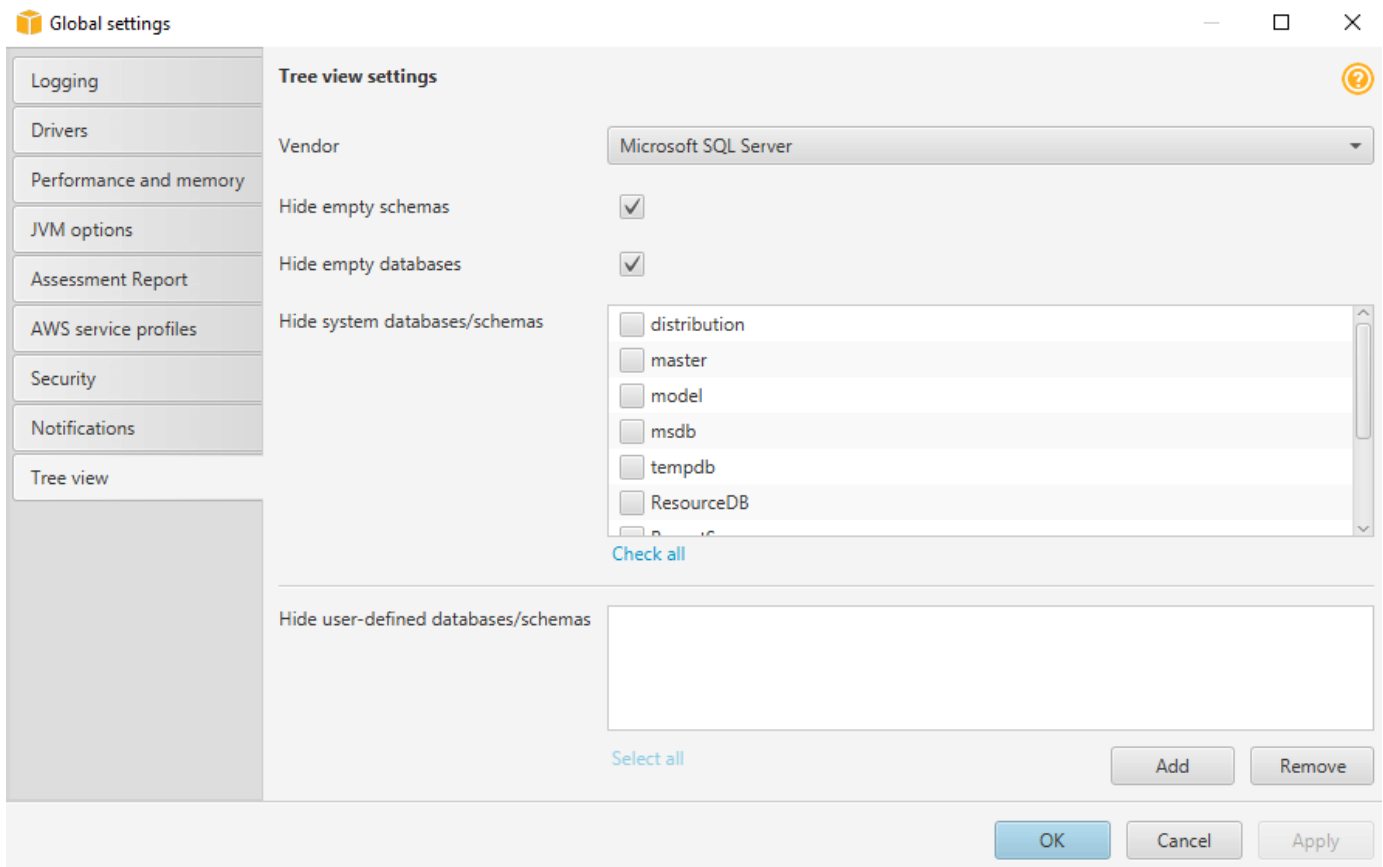
- `filterGroupType` è il tipo di regola di filtro (AND o operatori OR logici) che si applica a più clausole di filtro.
- `filterCategory` è il livello del filtro (Categorie o Stati).
- `names` è l'elenco dei nomi degli oggetti che si applica al filtro Categorie.
- `filterCondition` è la condizione (LIKE o NOT LIKE) di filtro che si applica al filtro Categorie.
- `transformName` è il nome dello stato che si applica al filtro Stato.
- `value` è il valore in base al quale filtrare l'albero.
- `transformValue` è il valore del filtro (TRUE o FALSE) che si applica al filtro Stato.

Nascondere gli schemi nella vista AWS SCT ad albero

Utilizzando le impostazioni della visualizzazione ad albero, è necessario specificare quali schemi e database intendi visualizzare nella visualizzazione ad albero di AWS SCT. Puoi nascondere schemi vuoti, database vuoti, database di sistema, oltre a database e schemi definiti dall'utente.

Nascondere database e schemi nella visualizzazione ad albero

1. Apri un progetto AWS SCT.
2. Connettiti al datastore che intendi visualizzare nella visualizzazione ad albero.
3. Scegli Impostazioni, Impostazioni globali, Vista ad albero.



4. Nella sezione Impostazioni della visualizzazione ad albero completa le seguenti operazioni:
 - Per Vendor, scegli la piattaforma di database.
 - Scegliete Nascondi schemi vuoti per nascondere gli schemi vuoti per la piattaforma di database selezionata.
 - Scegli Nascondi database vuoti per nascondere i database vuoti per la piattaforma di database selezionata.
 - Per Nascondi database/schemi di sistema, scegli i database e gli schemi di sistema per nome per nasconderli.
 - In Nascondi database/schemi definiti dall'utente, inserisci i nomi dei database e degli schemi definiti dall'utente che desideri nascondere, quindi scegli Aggiungi. I nomi non fanno distinzione tra maiuscole e minuscole.
5. Scegli OK.

Creazione e revisione del rapporto di valutazione della migrazione del database

Il rapporto di valutazione della migrazione del database riassume tutte le azioni relative agli schemi che non possono essere convertiti automaticamente nel motore dell'istanza DB Amazon RDS di destinazione. Il report include inoltre le stime dello sforzo necessario per scrivere il codice equivalente per l'istanza database di destinazione.

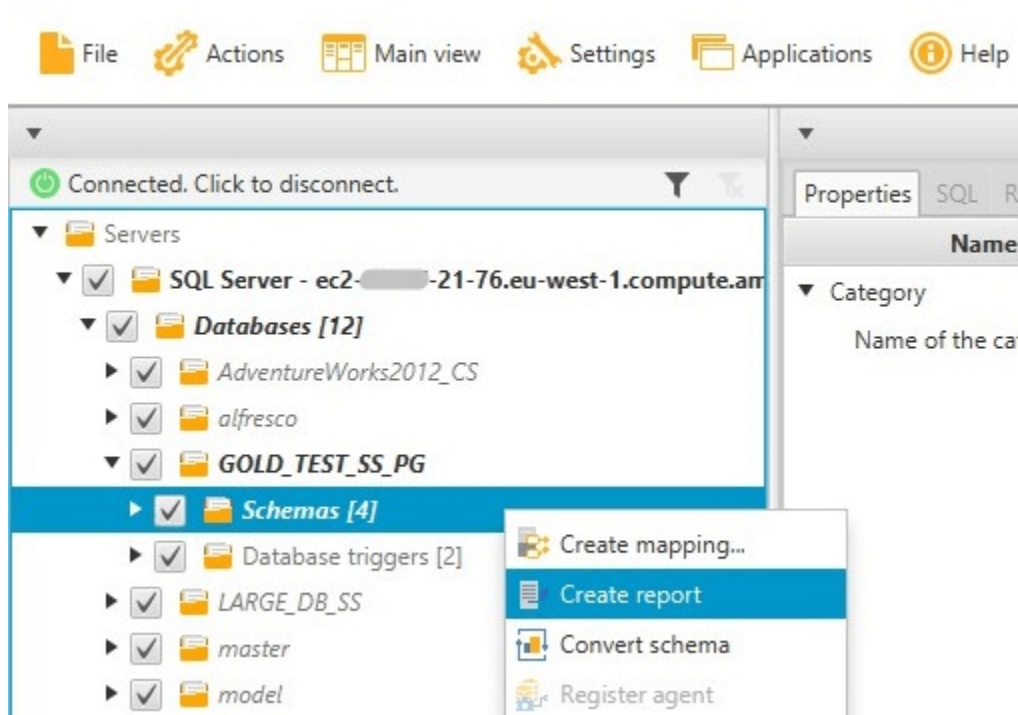
È possibile creare un rapporto di valutazione della migrazione del database dopo aver aggiunto i database di origine e le piattaforme di destinazione al progetto e aver specificato le regole di mappatura.

Creazione e visualizzazione del report di valutazione della migrazione del database

1. Assicurati di aver creato una regola di mappatura per lo schema del database di origine per cui creare un rapporto di valutazione. Per ulteriori informazioni, consulta [Aggiungere una nuova regola di mappatura](#).
2. Nel menu Visualizza, scegli Vista principale.
3. Nel pannello di sinistra che mostra lo schema dal database di origine, scegli gli oggetti dello schema per cui creare un rapporto di valutazione.

Assicurati di aver selezionato le caselle di controllo per tutti gli oggetti dello schema per cui creare un rapporto di valutazione.

4. Apri il menu di scelta rapida (tasto destro del mouse) per l'oggetto e quindi scegli Crea report.



Si aprirà la visualizzazione del report di valutazione.

5. Scegli la scheda Azioni.

La scheda Azioni mostra un elenco di elementi che descrivono lo schema che non può essere convertito automaticamente. Scegli una delle azioni nell'elenco. AWS SCT evidenzia l'elemento dello schema a cui si applica l'azione, come illustrato di seguito.

The screenshot shows the AWS Schema Conversion Tool interface. On the left, a tree view shows the project structure: Servers (SQL Server - ec2-...), Databases (12), and Schemas (1) including 'dbo' with 8 tables and 4 procedures. The main window displays a 'Summary' report with the following 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.

Below the issues, the SQL code for the procedure 'POSITION_UPDATE_CASH_CGT_BULK' is shown:

```

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

```

6. Seleziona la scheda Summary (Riepilogo).

Nella scheda Summary (Riepilogo) vengono visualizzate le informazioni di riepilogo del report di valutazione della migrazione del database. Visualizza il numero degli elementi convertiti automaticamente e il numero di quelli non convertiti automaticamente. Il riepilogo include inoltre una stima del tempo necessario per creare schemi nella istanza database di destinazione che siano equivalenti a quelli presenti nel database di origine.

La sezione Valutazione delle licenze e Support cloud contiene informazioni sullo spostamento dello schema di database locale esistente su un'istanza DB Amazon RDS che esegue lo stesso motore. Ad esempio, se si desidera modificare i tipi di licenza, questa sezione del rapporto indica quali funzionalità dal database corrente rimuovere.

È riportato di seguito un riepilogo di report di valutazione.

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 conversions.

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%	33%	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%	0%	80%	0%

7. Seleziona la scheda Summary (Riepilogo), quindi seleziona Save to PDF (Salva in PDF). Il report di valutazione della migrazione del database viene salvato come file PDF. Il file PDF contiene sia il riepilogo, sia le informazioni sulle attività.

Puoi anche scegliere Salva in CSV per salvare il rapporto come file CSV. Quando scegli questa opzione, AWS SCT crea tre file CSV. Questi file contengono le seguenti informazioni:

- Un elenco di azioni di conversione con azioni consigliate.
- Un riepilogo delle azioni di conversione con una stima dello sforzo richiesto per convertire un'occorrenza dell'azione.
- Un riepilogo con una serie di azioni classificate in base al tempo stimato di conversione.

Database objects with conversion actions for Amazon RDS for PostgreSQL

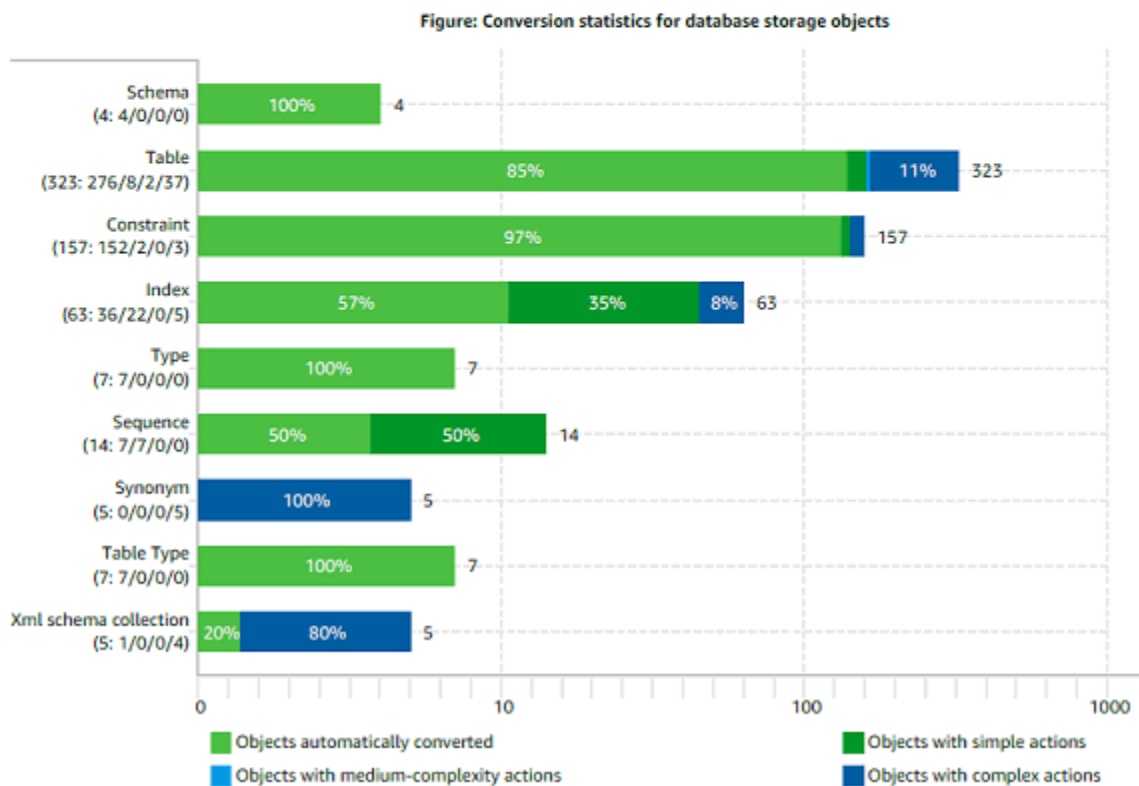
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

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."

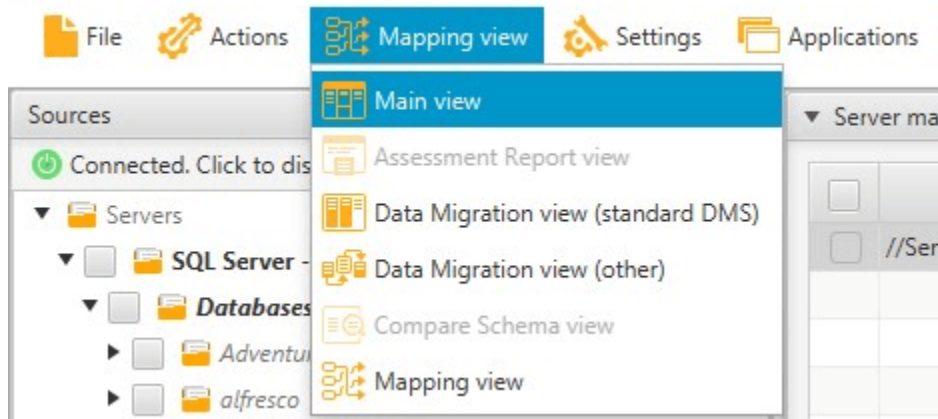


Conversione dello schema

Dopo aver aggiunto i database di origine e di destinazione al progetto e aver creato le regole di mappatura, è possibile convertire gli schemi del database di origine. Utilizza la procedura seguente per convertire lo schema.

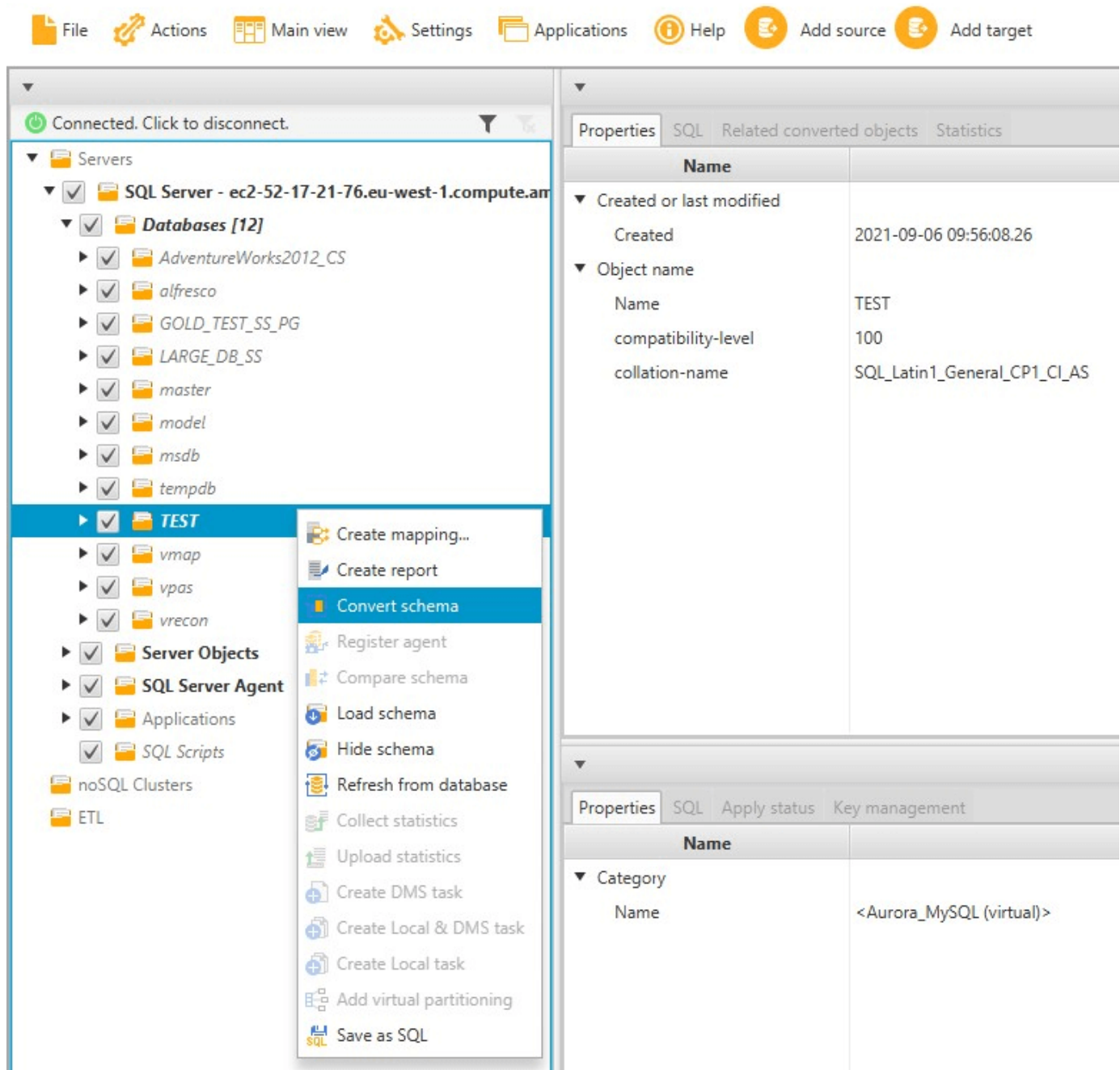
Per convertire lo schema

1. Scegliete Visualizza, quindi scegliete Vista principale.



2. Nel pannello di sinistra che mostra lo schema dal database di origine, seleziona la casella di controllo per il nome dell'oggetto da convertire. Quindi, scegli questo oggetto. AWS SCT evidenzia il nome dell'oggetto in blu. Apri il menu di scelta rapida (tasto destro del mouse) per l'oggetto e scegli Converti schema.

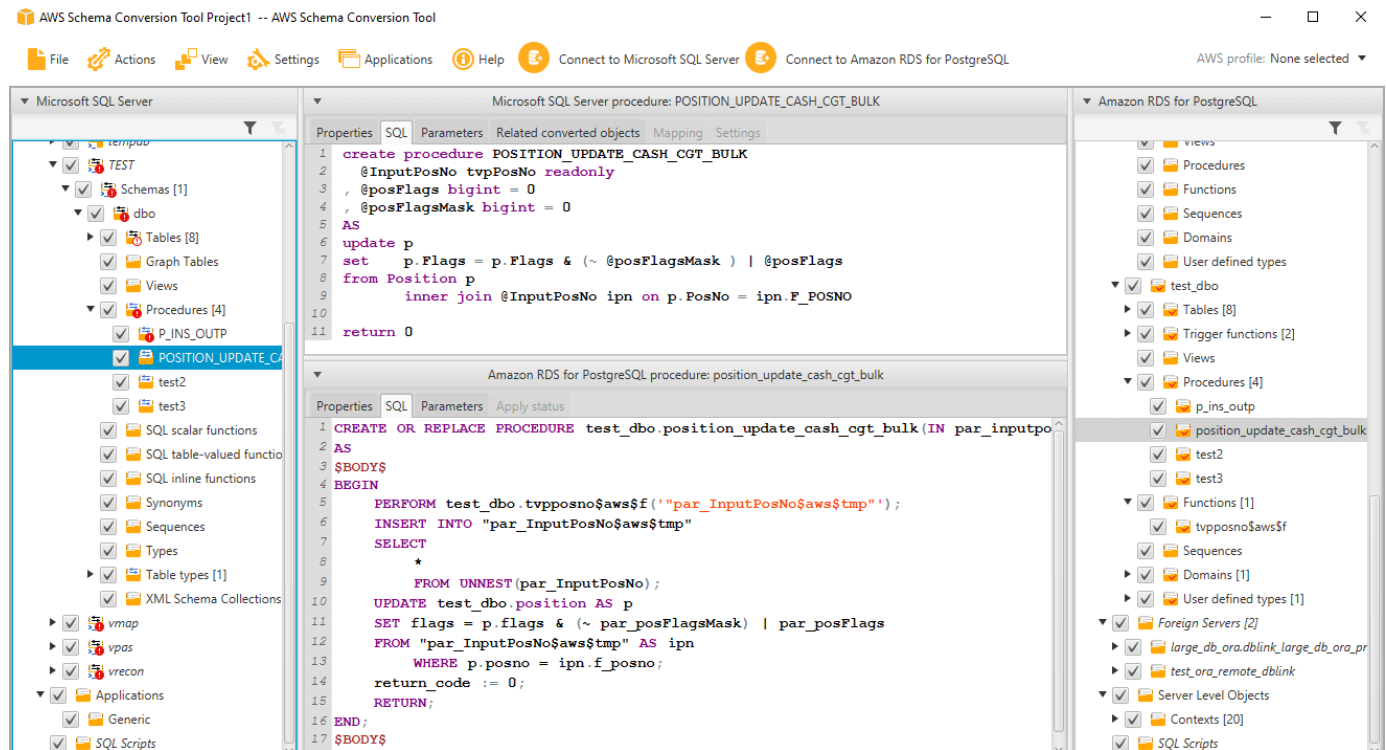
Per convertire diversi oggetti del database, selezionare le caselle di controllo relative a tutti gli oggetti. A questo punto, seleziona il nodo principale. Ad esempio, per le tabelle, il nodo padre è Tables. Assicurati che AWS SCT evidenzi il nome del nodo principale in blu. Apri il menu di scelta rapida (tasto destro del mouse) per il nodo principale e scegli Converti schema.



- Quando AWS SCT termina la conversione dello schema, puoi visualizzare la proposta di schema nel pannello a destra del progetto.

A questo punto, nessuno schema viene applicato all'istanza del database di destinazione. Lo schema pianificato è parte del tuo progetto. Se si sceglie un elemento dello schema convertito, è possibile visualizzare il comando dello schema pianificato nel pannello in basso al centro dell'istanza del database di destinazione.

Puoi modificare lo schema in questa finestra. Lo schema modificato viene archiviato come parte del progetto e viene scritto nell'istanza del database di destinazione quando si sceglie di applicare lo schema convertito.



Applicazione dello schema convertito all'istanza database di destinazione

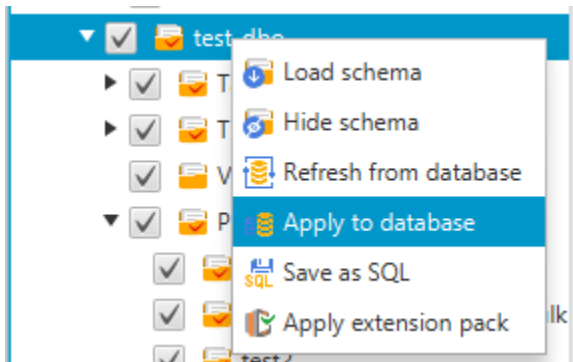
Puoi applicare lo schema del database convertito alla tua istanza database di destinazione. Dopo che lo schema è stato applicato alla tua istanza database di destinazione, puoi aggiornare lo schema in base alle attività nel report di valutazione della migrazione del database.

Warning

La procedura seguente sovrascrive lo schema di destinazione esistente. Fare attenzione a non sovrascrivere gli schemi involontariamente. Fai attenzione a non sovrascrivere gli schemi nell'istanza DB di destinazione che hai già modificato o a sovrascrivere tali modifiche.

Per applicare lo schema del database convertito all'istanza del database di destinazione

1. Scegli Connect al server nella parte superiore del pannello destro del progetto per connetterti al database di destinazione. Se sei connesso al tuo database di destinazione, salta questo passaggio.
2. Scegli l'elemento dello schema nel pannello destro del tuo progetto che mostra lo schema pianificato per la tua istanza database di destinazione;
3. Apri il menu contestuale (tasto destro del mouse) per l'elemento dello schema e quindi scegli Apply to database (Applica al database).



Lo schema convertito è applicato all'istanza database di destinazione.

Archiviazione dei profili AWS di servizio in AWS SCT

Puoi archiviare AWS le tue credenziali in AWS SCT. AWS SCT utilizza le tue credenziali quando utilizzi funzionalità che si integrano con AWS i servizi. Ad esempio, AWS SCT si integra con Amazon S3, AWS Lambda, Amazon Relational Database Service (Amazon RDS) e AWS Database Migration Service (AWS DMS).

AWS SCT ti chiede AWS le tue credenziali quando accedi a una funzionalità che le richiede. Puoi archiviare le credenziali nelle impostazioni globali dell'applicazione. Quando AWS SCT richiede le credenziali, puoi selezionare le credenziali archiviate.

È possibile memorizzare diversi set di AWS credenziali nelle impostazioni globali dell'applicazione. Ad esempio, puoi archiviare un set di credenziali utilizzate in scenari di test e un altro set di credenziali utilizzate in scenari di produzione. Puoi anche memorizzare credenziali diverse per diversi Regione AWS s.

ArchiviazioneAWS delle credenziali

Utilizza la seguente procedura per archiviareAWS le credenziali a livello globale.

Per archiviareAWS le credenziali

1. Avvia il AWS Schema Conversion Tool.
2. Apri il menu Impostazioni, quindi scegli Impostazioni globali. Si aprirà la finestra di dialogo Global Settings (Impostazioni globali).
3. Scegli i profili diAWS servizio, quindi scegli Aggiungi un nuovo profiloAWS di servizio.
4. Inserisci le tueAWS informazioni come segue.

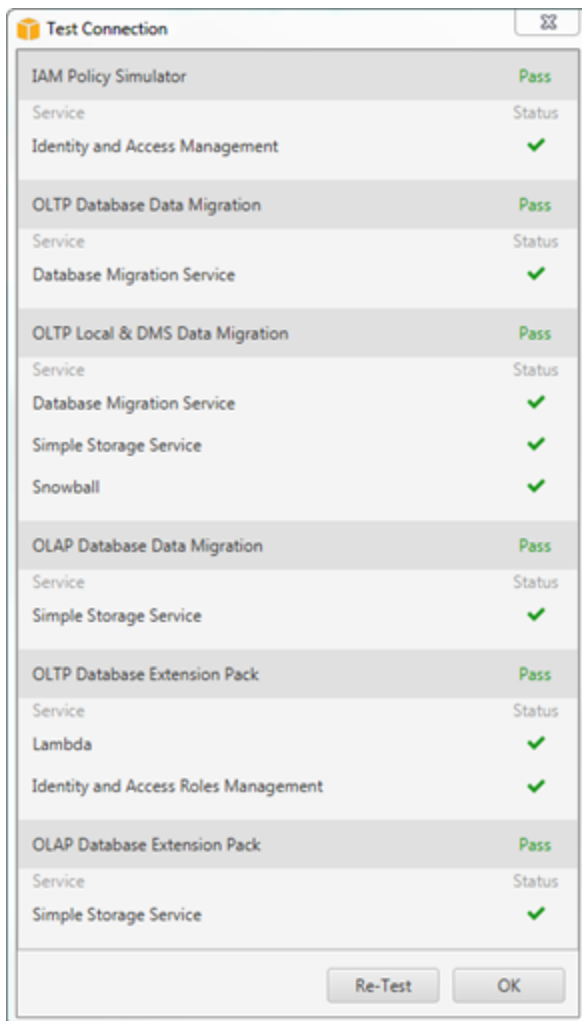
Opzione AWS SCT	Operazione
Profile name (Nome profilo)	Inserisci un nome per il tuo profilo.
AWSchiave di accesso	Inserisci la tua chiaveAWS d'accesso.
AWSchiave segreta	Inserisci la chiave di accessoAWS segreta. Per ulteriori informazioni sulle chiavi diAWS accesso, consulta Gestione delle chiavi di accesso nella Guida per l'utente di IAM.
Regione	Scegli ilRegione AWS per il tuo profilo.
Cartella bucket Amazon S3	Scegli il bucket Amazon S3 per il tuo profilo. È necessario specificare un bucket solo se si utilizza una funzionalità che si connette ad Amazon S3. Per ulteriori informazioni sui privilegi richiesti, consulta Autorizzazioni per l'utilizzo del profiloAWS di servizio .

Seleziona Use FIPS endpoint for S3 (Utilizza endpoint FIPS per S3) se hai bisogno di rispettare i requisiti di sicurezza dello standard FIPS (Federal Information Processing Standard). Gli endpoint FIPS sono disponibili nelle seguentiAWS regioni:

- US East (N. Virginia) Region
- US East (Ohio) Region

- US West (N. California) Region
 - US West (Oregon) Region
5. Scegli Verifica connessione per verificare che le tue credenziali siano corrette e attive.

Viene visualizzata la finestra di dialogo Verifica connessione. Puoi visualizzare lo stato di ciascuno dei servizi connessi al profilo. Pass (Trasmetti) indica che il profilo è in grado di accedere correttamente al servizio.



6. Dopo aver configurato il profilo, seleziona Save (Salva) per salvare il profilo o Cancel (Annulla) per annullare le modifiche.
7. Scegliete OK per chiudere la finestra di dialogo Impostazioni globali.

Impostazione del profilo predefinito per un progetto

Puoi impostare il profilo predefinito di un progetto AWS SCT. In questo modo le AWS credenziali memorizzate nel profilo vengono associate al progetto. Con il progetto aperto, utilizza la procedura seguente per impostare il profilo predefinito.

Impostazione del profilo predefinito di un progetto

1. Iniziare il progetto AWS Schema Conversion Tool e creare un nuovo progetto.
2. Nel menu Impostazioni, scegli Impostazioni progetto. Viene visualizzata la finestra di dialogo Impostazioni progetto.
3. Scegli la scheda Ambiente del progetto.
4. Scegli Aggiungi un nuovo profilo AWS di servizio per aggiungere un nuovo profilo. Quindi, per il profilo di AWS servizio, scegli il profilo che desideri associare al progetto.
5. Scegliete OK per chiudere la finestra di dialogo delle impostazioni del progetto. Puoi inoltre selezionare Cancel (Annulla) per annullare le modifiche.

Autorizzazioni per l'utilizzo del profilo AWS di servizio

Le seguenti autorizzazioni sono necessarie per accedere al bucket Amazon S3 dal tuo profilo AWS di servizio:

- `s3:PutObject`— per aggiungere oggetti nel bucket Amazon S3.
- `s3:DeleteObject`— per rimuovere la versione nulla di un oggetto e inserire un indicatore di eliminazione, che diventa la versione corrente dell'oggetto.
- `s3:ListBucket`— per restituire fino a 1.000 oggetti dal tuo bucket Amazon S3.
- `s3:GetObject`— per recuperare oggetti dal bucket Amazon S3.

I seguenti esempi di codice mostrano come concedere queste autorizzazioni all'utente.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:DeleteObject",
```

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

Utilizzo di AWS Secrets Manager

AWS SCT può utilizzare le credenziali del database in cui sono archiviate AWS Secrets Manager. È possibile inserire tutti i valori nella finestra di dialogo di connessione al database da Secrets Manager. Per utilizzare Secrets Manager, assicurati di archiviare AWS i profili in AWS Schema Conversion Tool.

Per ulteriori informazioni sull'utilizzo AWS Secrets Manager, vedi [Cos'è AWS Secrets Manager?](#) nella Guida per l'AWS Secrets Manager utente. Per ulteriori informazioni sulla memorizzazione dei AWS profili, consulta [Archiviazione dei profili AWS di servizio in AWS SCT](#).

Per recuperare le credenziali del database da Secrets Manager

1. Iniziare il progetto AWS Schema Conversion Tool e creare un nuovo progetto.
2. Scegli Aggiungi fonte o Aggiungi destinazione per aggiungere un nuovo database al tuo progetto.
3. Scegli una piattaforma di database, quindi scegli Avanti.
4. Per AWS Secret, scegli il segreto che desideri utilizzare.
5. Scegli Popola. Quindi AWS SCT inserisce tutti i valori nella finestra di dialogo di connessione al database.
6. Scegli Verifica connessione per verificare che sia in AWS SCT grado di connettersi al tuo database.
7. Scegli Connect per connetterti al tuo database.

AWS SCT supporta segreti che hanno la seguente struttura.

```
{
```

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

In questa struttura, `password` `username` and sono obbligatori e tutti gli altri valori sono opzionali. Assicurati che i valori archiviati in Secrets Manager includano tutte le credenziali del database.

Memorizzare le password del database

Puoi archiviare una password o un certificato SSL di un database nella cache di AWS SCT. Per archiviare una password, seleziona Store Password (Archivia password) al momento della creazione di una connessione.

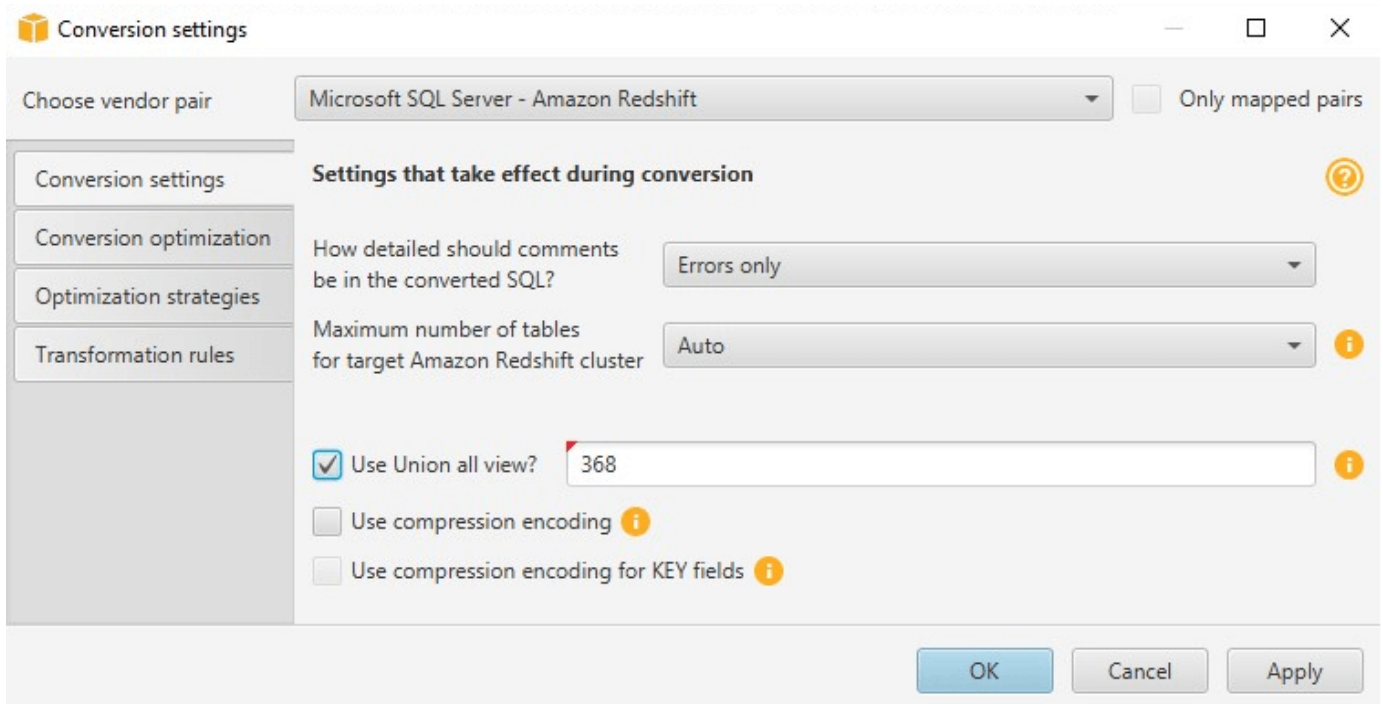
La password viene crittografata utilizzando il token generato in modo casuale nel file `seed.dat`. La password è quindi archiviata con il nome utente nel file di cache. Se smarrisci il file `seed.dat`, o questo risulta danneggiato, la password del database potrebbe essere crittografata in modo errato. In tal caso, la connessione avrà esito negativo.

Utilizzo della vista UNION ALL per progetti con tabelle partizionate

Se una tabella di origine è partizionata, AWS SCT crea `n` tabelle di destinazione, dove `n` è il numero di partizioni nella tabella di origine. AWS SCT crea una vista UNION ALL sopra le tabelle di destinazione per rappresentare la tabella di origine. Se si utilizza un estrattore di AWS SCT dati per migrare i dati, le partizioni della tabella di origine verranno estratte e caricate in parallel da sottoattività separate.

Per usare la vista Union All per un progetto

1. Avvia AWS SCT. Crea un nuovo progetto o apri un AWS SCT progetto esistente.
2. Nel menu Impostazioni, scegli Impostazioni di conversione.
3. Scegli una coppia di database OLAP dall'elenco in alto.
4. Attivare Use Union all view?



5. Scegliete OK per salvare le impostazioni e chiudere la finestra di dialogo delle impostazioni di conversione.

Scelte rapide da tastiera per AWS SCT

Di seguito sono elencate le scorciatoie da tastiera che è possibile utilizzare AWS SCT.

Tasto di scelta rapida	Descrizione
Ctrl+N	Crea un nuovo progetto.
Ctrl+O	Apri un progetto esistente.
Ctrl+S	Salva un progetto aperto.
Ctrl+W	Crea un nuovo progetto utilizzando la procedura guidata.
CTRL+M	Crea una nuova valutazione multiserver.
Ctrl+L	Aggiungere un nuovo database di origine.
Ctrl+R	Aggiungere un nuovo database di destinazione.

Tasto di scelta rapida	Descrizione
CTRL+F4	Chiudi un progetto aperto.
F1	Apri la Guida perAWS SCT l'utente.

Nozioni di base su AWS SCT

Puoi utilizzare il plugin AWS Schema Conversion Tool (AWS SCT) per convertire lo schema per un database di origine. Il database di origine può essere un motore autogestito in esecuzione in locale o su un'istanza Amazon EC2. È possibile convertire lo schema di origine in uno schema per qualsiasi database supportato ospitato da AWS. L'applicazione AWS SCT fornisce un'interfaccia utente basata su progetto.

La maggior parte del lavoro che svolgi con AWS SCT inizia nel seguente modo:

1. Installare AWS SCT. Per ulteriori informazioni, consulta la pagina [Installazione, verifica e aggiornamento AWS SCT](#).
2. Installazione di AWS SCT agente, se necessario. AWS SCT gli agenti sono necessari solo per determinati scenari di migrazione, ad esempio tra origini e destinazioni eterogenee. Per ulteriori informazioni, consulta la pagina [Migrazione dei dati da un data warehouse locale ad Amazon Redshift](#).
3. Acquisisci familiarità con l'interfaccia utente di AWS SCT. Per ulteriori informazioni, consulta la pagina [Utilizzo dell'interfaccia AWS SCT utente](#).
4. Creazione di un progetto AWS SCT. Esegui la connessione ai tuoi database di origine e di destinazione. Per ulteriori informazioni sulla connessione al tuo database di origine, consulta [Origini per AWS SCT](#).
5. Crea regole di mappatura. Per ulteriori informazioni sulle regole di mappatura, consulta [Creazione di regole di mappatura in AWS SCT](#).
6. Esegui e quindi esamina il report di valutazione della migrazione database. Per ulteriori informazioni sul report di valutazione, consulta [Creazione e revisione del rapporto di valutazione della migrazione del database](#).
7. Converti gli schemi del database di origine. Sono diversi gli aspetti della conversione che devi tenere a mente, come la gestione delle voci che non vengono convertite e le modalità di mappatura delle voci che devono essere convertite in un modo particolare. Per ulteriori informazioni sulla conversione di uno schema di origine, consulta [Conversione di schemi di database utilizzando AWS SCT](#).

Anche nel caso della conversione dello schema di un data warehouse devi prendere in considerazione alcuni aspetti prima di procedere. Per ulteriori informazioni, consulta la pagina [Conversione di schemi di data warehouse in Amazon Redshift utilizzando AWS SCT](#).

8. Applica la conversione dello schema alla tua destinazione. Per ulteriori informazioni sull'applicazione della conversione di uno schema di origine, consulta [Applicazione del codice convertito](#).
9. Puoi anche utilizzare AWS SCT per convertire le procedure SQL archiviate e altro codice dell'applicazione. Per ulteriori informazioni, consulta la pagina [Conversione del codice SQL dell'applicazione utilizzando AWS SCT](#)

Inoltre, puoi utilizzare AWS SCT per migrare i dati da un database di origine a un database gestito da Amazon. Per alcuni esempi, consulta [Migrazione dei dati da un data warehouse locale ad Amazon Redshift](#).

Origini per AWS SCT

AWS Schema Conversion Tool (AWS SCT) può convertire gli schemi dai seguenti database di origine e data warehouse in un database o data warehouse di destinazione. Per informazioni su autorizzazioni, connessioni e cosa AWS SCT può essere convertito per l'uso con il database o il data warehouse di destinazione, consulta i dettagli nei seguenti argomenti.

Informazioni sulla crittografia

[Crittografia delle connessioni Amazon RDS](#)

Fonti del database

- [Usare Apache Cassandra come fonte](#)
- [Usare il database SQL di Azure come origine](#)
- [Utilizzo di IBM Db2 for z/OS come fonte](#)
- [Utilizzo di IBM Db2 LUW come origine](#)
- [Utilizzo di MySQL come origine](#)
- [Utilizzo del database Oracle come fonte](#)
- [Utilizzo di PostgreSQL come origine](#)
- [Utilizzo di SAP ASE \(Sybase ASE\) come fonte](#)
- [Utilizzo di SQL Server come origine](#)

Fonti di data warehouse

- [Utilizzo di Amazon Redshift come fonte](#)
- [Usare Azure Synapse Analytics come fonte](#)
- [Utilizzo di BigQuery come origine](#)
- [Utilizzo del database Greenplum come fonte](#)
- [Usare Netezza come fonte](#)
- [Utilizzo di Oracle Data Warehouse come origine](#)
- [Usare Snowflake come fonte](#)
- [Utilizzo di SQL Server Data Warehouse come origine](#)

- [Utilizzo di Teradata come fonte](#)
- [Usare Vertica come fonte](#)

Grandi fonti di dati

- [Usare Apache Hadoop come fonte](#)
- [Usare Apache Oozie come fonte](#)

Crittografia delle connessioni Amazon RDS e Amazon Aurora in AWS SCT

Per aprire connessioni crittografate ai database Amazon RDS o Amazon Aurora da un'applicazione, devi importare i certificati AWS root in una qualche forma di storage delle chiavi. Puoi scaricare i certificati root dalla AWS pagina [Utilizzo di SSL/TLS per crittografare una connessione a un'istanza database nella Guida per l'utente di Amazon RDS](#).

Sono disponibili due opzioni: un certificato root che funziona per tutte le AWS regioni e un pacchetto di certificati che contiene sia i vecchi che i nuovi certificati root.

A seconda del dispositivo che desideri utilizzare, segui i passaggi di una delle due procedure seguenti.

Per importare il certificato o i certificati nella memoria di sistema di Windows

1. Scarica uno o più certificati da una delle seguenti fonti:

Per informazioni sul download dei certificati, consulta [Utilizzo di SSL/TLS per crittografare una connessione a un'istanza database nella Guida per l'utente di Amazon RDS](#).

2. Nella finestra di ricerca di Windows, inserisci **Manage computer certificates**. Quando viene richiesto se consentire all'applicazione di apportare modifiche al computer, scegli Sì.
3. Quando si apre la finestra dei certificati, se necessario, espandi Certificati - Computer locale in modo da poter visualizzare l'elenco dei certificati. Apri il menu contestuale (clic con il pulsante destro del mouse) per Trusted Root Certification Authorities, quindi scegli Tutte le attività, Importa.
4. Scegli Avanti, quindi Sfoglia e trova il *.pem file scaricato nel passaggio 1. Scegliete Apri per selezionare il file del certificato, scegliete Avanti, quindi scegliete Fine.

Note

Per trovare il file, modifica il tipo di file nella finestra di ricerca su Tutti i file (*.*), poiché non .pem è un'estensione del certificato standard.

5. Nella Microsoft Management Console, espandi Certificati. Quindi espandi Trusted Root Certification Authorities, scegli Certificati e trova il certificato per confermarne l'esistenza. Il nome del certificato inizia con Amazon RDS.
6. Riavviare il computer.

Per importare il certificato o i certificati in Java KeyStore

1. Scarica il certificato o i certificati da una delle seguenti fonti:

Per informazioni sul download dei certificati, consulta [Utilizzo di SSL/TLS per crittografare una connessione a un'istanza database nella Guida per l'utente di Amazon RDS](#).

2. Se hai scaricato il pacchetto di certificati, dividilo in singoli file di certificati. A tale scopo, inserisci ogni blocco di certificato, che inizia con -----BEGIN CERTIFICATE----- e termina con, -----END CERTIFICATE----- in un *.pem file separato. Dopo aver creato un *.pem file separato per ogni certificato, puoi rimuovere in sicurezza il file del pacchetto di certificati.
3. Apri una finestra di comando o una sessione di terminale nella directory in cui hai scaricato il certificato ed esegui il comando seguente per ogni *.pem file creato nel passaggio precedente.

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

Example

L'esempio seguente presuppone che il eu-west-1-bundle.pem file sia stato scaricato.

```
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. Aggiungi il keystore come archivio fiduciario in AWS SCT. Per farlo, dal menu principale scegli Impostazioni, Impostazioni globali, Sicurezza, Archivio attendibile, quindi scegli Seleziona archivio attendibile esistente.

Dopo aver aggiunto il trust store, puoi utilizzarlo per configurare una connessione SSL abilitata quando crei una AWS SCT connessione al database. Nella finestra di dialogo AWS SCT Connetti al database, scegli Usa SSL e scegli il trust store inserito in precedenza.

Utilizzo di Apache Cassandra come fonte per AWS SCT

Puoi utilizzare AWS SCT per convertire i keyspaces da Apache Cassandra ad Amazon DynamoDB.

Connessione ad Apache Cassandra come fonte

Usa la seguente procedura per connetterti al tuo database di origine di Apache Cassandra con AWS Schema Conversion Tool

Per connettersi a un database sorgente di Apache Cassandra

1. Nel AWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Cassandra, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per AWS Secret, scegli il nome del segreto.
 2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine di Apache Cassandra, utilizzate le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Immetti il nome Domain Name Service (DNS) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
User name (Nome utente) e Password (Password)	<p>Inserisci le credenziali del database per connetterti al server del database di origine.</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p>
Use SSL (Usa SSL)	<p>Scegli questa opzione se desideri utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none"> • Trust store: il trust store da utilizzare. • Archivio chiavi: l'archivio chiavi da utilizzare.
Memorizza la password	<p>AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza dover inserire la password.</p>

5. Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
6. Scegli Connetti per connetterti al tuo database di origine.

Utilizzo di Apache Hadoop come fonte per AWS SCT

Puoi utilizzare l'interfaccia a riga di AWS SCT comando (CLI) per migrare da Apache Hadoop ad Amazon EMR. AWS SCT utilizza il tuo bucket Amazon S3 come spazio di archiviazione temporaneo per i tuoi dati durante la migrazione.

AWS SCT supporta come sorgente Apache Hadoop versione 2.2.0 e successive. Inoltre, AWS SCT supporta Apache Hive versione 0.13.0 e successive.

AWS SCT supporta come destinazione Amazon EMR versione 6.3.0 e successive. Inoltre, AWS SCT supporta come destinazione Apache Hadoop versione 2.6.0 e successive e Apache Hive versione 0.13.0 e successive.

Argomenti

- [Prerequisiti per l'utilizzo di Apache Hadoop come sorgente](#)
- [Autorizzazioni per l'utilizzo di Hive come fonte](#)
- [Autorizzazioni per l'utilizzo di HDFS come fonte](#)
- [Autorizzazioni per l'utilizzo di HDFS come destinazione](#)
- [Connessione ad Apache Hadoop come fonte](#)
- [Connessione ai servizi Hive e HDFS di origine](#)
- [Connessione ad Amazon EMR come destinazione](#)

Prerequisiti per l'utilizzo di Apache Hadoop come sorgente

I seguenti prerequisiti sono necessari per connettersi ad Apache Hadoop con la CLI. AWS SCT

- Crea un bucket Amazon S3 per archiviare i dati durante la migrazione. Puoi quindi copiare i dati su Amazon EMR HDFS o utilizzare Amazon S3 come archivio di dati per i tuoi carichi di lavoro Hadoop. Per ulteriori informazioni, consulta [Creazione di un bucket](#) nella Guida per l'utente di Amazon S3.
- Crea un ruolo AWS Identity and Access Management (IAM) con la AmazonS3FullAccess policy. AWS SCT utilizza questo ruolo IAM per accedere al tuo bucket Amazon S3.
- Prendi nota della tua chiave AWS segreta e della tua chiave di accesso AWS segreta. Per ulteriori informazioni sulle chiavi di AWS accesso, consulta [Managing Access Keys](#) nella IAM User Guide.
- Crea e configura un cluster Amazon EMR di destinazione. Per ulteriori informazioni, consulta la [Guida introduttiva ad Amazon EMR](#) nella Guida alla gestione di Amazon EMR.

- Installa l'`distcp` utilità sul tuo cluster Apache Hadoop di origine. Inoltre, installa l'`s3-dist-cp` utilità sul cluster Amazon EMR di destinazione. Assicurati che gli utenti del tuo database dispongano delle autorizzazioni necessarie per eseguire queste utilità.
- Configura il `core-site.xml` file nel tuo cluster Hadoop di origine per utilizzare il protocollo s3a. A tale scopo, impostate il `fs.s3a.aws.credentials.provider` parametro su uno dei seguenti valori.
 - `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`

È possibile aggiungere il seguente esempio di codice nel `core-site.xml` file.

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

L'esempio precedente mostra una delle quattro opzioni del precedente elenco di opzioni. Se non imposti il `fs.s3a.aws.credentials.provider` parametro nel `core-site.xml` file, AWS SCT sceglie automaticamente il provider.

Autorizzazioni per l'utilizzo di Hive come fonte

Le autorizzazioni richieste per un utente sorgente di Hive sono le seguenti:

- **READ**accesso alle cartelle dei dati di origine e al bucket Amazon S3 di origine
- **READ+WRITE**accesso ai bucket Amazon S3 intermedi e di destinazione

Per aumentare la velocità di migrazione, ti consigliamo di eseguire la compattazione per le tabelle di origine transazionali ACID.

Le autorizzazioni richieste per un utente target di Amazon EMR Hive sono le seguenti:

- **READ**accesso al bucket Amazon S3 di destinazione
- **READ+WRITE**accesso al bucket Amazon S3 intermedio

- READ+WRITEaccesso alle cartelle HDFS di destinazione

Autorizzazioni per l'utilizzo di HDFS come fonte

Le autorizzazioni richieste per HDFS come sorgente sono le seguenti:

- EXECUTE per NameNode
- EXECUTE+READper tutte le cartelle e i file di origine inclusi nel progetto di migrazione
- READ+WRITEper la tmp directory in cui NameNode eseguire i job Spark e archiviare i file prima della migrazione ad Amazon S3

In HDFS, tutte le operazioni richiedono l'accesso trasversale. L'accesso trasversale richiede l'EXECUTEautorizzazione per tutti i componenti esistenti del percorso, ad eccezione del componente finale del percorso. Ad esempio, per qualsiasi operazione di accesso/fo`o`/ba`r`/ba`z`, l'utente deve disporre dell'EXECUTEautorizzazione su //fo`o`, e/fo`o`/ba`r`.

L'esempio di codice seguente mostra come concedere EXECUTE+READ le autorizzazioni per le cartelle e i file di origine e le READ+WRITE autorizzazioni per la directory. tmp

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

Autorizzazioni per l'utilizzo di HDFS come destinazione

Le autorizzazioni richieste per Amazon EMR HDFS come destinazione sono le seguenti:

- EXECUTEper il NameNode cluster Amazon EMR di destinazione
- READ+WRITEper le cartelle HDFS di destinazione in cui archiverai i dati dopo la migrazione

Connessione ad Apache Hadoop come fonte

È possibile utilizzare Apache Hadoop come sorgente nella AWS SCT versione 1.0.670 o superiore. Puoi migrare i cluster Hadoop su Amazon EMR solo nell'interfaccia a riga di AWS SCT comando (CLI). Prima di iniziare, acquisisci familiarità con l'interfaccia a riga di comando di AWS SCT. Per ulteriori informazioni, consulta [AWS SCT Riferimento CLI](#).

Per connettersi ad Apache Hadoop nella CLI AWS SCT

1. Crea un nuovo script AWS SCT CLI o modifica un modello di scenario esistente. Ad esempio, puoi scaricare e modificare il `HadoopMigrationTemplate.scts` modello. Per ulteriori informazioni, consulta [Ottenere scenari CLI](#).
2. Configura le impostazioni AWS SCT dell'applicazione come la posizione del driver e la cartella di registro.

Scarica il driver JDBC richiesto e specifica la posizione in cui archiviare il file. Per ulteriori informazioni, consulta [Scaricamento dei driver di database richiesti](#).

Il seguente esempio di codice mostra come aggiungere il percorso al driver Apache Hive. Dopo aver eseguito questo esempio di codice, AWS SCT memorizza i file di registro nella `c:\sct` cartella.

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

È possibile utilizzare questo esempio e i seguenti in Windows.

3. Crea un nuovo AWS SCT progetto.

L'esempio di codice seguente crea il `hadoop_emr` progetto nella `c:\sct` cartella.

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

4. Aggiungi il tuo cluster Hadoop di origine al progetto.

Usa il `AddSourceCluster` comando per connetterti al cluster Hadoop di origine. Assicurati di fornire valori per i seguenti parametri obbligatori: `namehost`, `port`, `euser`. Gli altri parametri sono opzionali.

L'esempio di codice seguente aggiunge il cluster Hadoop di origine. Questo esempio viene impostato HADOOP_SOURCE come nome del cluster di origine. Utilizzate questo nome di oggetto per aggiungere servizi Hive e HDFS al progetto e creare regole di mappatura.

```
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'
/
```

Nell'esempio precedente, sostituisci *hadoop_address* con l'indirizzo IP del tuo cluster Hadoop. Se necessario, configura il valore dell'opzione porta. Quindi, sostituisci *hadoop_user* e *hadoop_password* con il nome del tuo utente Hadoop e la password per questo utente. In *path* \ *name*, inserisci il nome e il percorso del file PEM per il cluster Hadoop di origine.

5. Salva il tuo script CLI. Quindi, aggiungi le informazioni di connessione per i tuoi servizi Hive e HDFS.

Connessione ai servizi Hive e HDFS di origine

Puoi connetterti ai servizi Hive e HDFS di origine con la AWS SCT CLI. Per connetterti ad Apache Hive, usa il driver Hive JDBC versione 2.3.4 o superiore. Per ulteriori informazioni, consulta [Scaricamento dei driver di database richiesti](#).

AWS SCT si connette ad Apache Hive con l'utente del hadoop cluster. A tale scopo, utilizzate i `AddSourceClusterHDFS` e `AddSourceClusterHive` comandi. È possibile utilizzare uno dei seguenti approcci.

- Crea un nuovo tunnel SSH.

In `createTunnel`, immettere **true**. Per `host`, inserisci l'indirizzo IP interno del tuo servizio Hive o HDFS di origine. Per `port`, inserisci la porta di servizio del tuo servizio Hive o HDFS.

Quindi, inserisci le tue credenziali Hive o HDFS per e. `user password` Per ulteriori informazioni sui tunnel SSH, consulta [Configurare un tunnel SSH verso il nodo primario utilizzando il port forwarding locale](#) nella Amazon EMR Management Guide.

- Usa un tunnel SSH esistente.

In `host`, immettere **localhost**. Per `port`, inserisci la porta locale dai parametri del tunnel SSH.

- Connettiti direttamente ai tuoi servizi Hive e HDFS.

Per `host`, inserisci l'indirizzo IP o il nome host del servizio Hive o HDFS di origine. Per `port`, inserisci la porta di servizio del tuo servizio Hive o HDFS. Quindi, inserisci le tue credenziali Hive o HDFS per e. `user password`

Per connettersi a Hive e HDFS nella CLI AWS SCT

1. Apri il tuo script CLI che include le informazioni di connessione per il tuo cluster Hadoop di origine. Assicurati di utilizzare il nome del cluster Hadoop definito nel passaggio precedente.
2. Aggiungi il tuo servizio Hive di origine al progetto.

Usa il `AddSourceClusterHive` comando per connettere il servizio Hive di origine. Assicurati di fornire valori per i seguenti parametri obbligatori: `userpassword`, `cluster`, `name`, e `port`. Gli altri parametri sono opzionali.

L'esempio di codice seguente crea un tunnel AWS SCT per funzionare con il tuo servizio Hive. Questo servizio Hive di origine viene eseguito sullo stesso PC di AWS SCT. Questo esempio utilizza il cluster di `HADOOP_SOURCE` origine dell'esempio precedente.

```
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'
```

/

Il seguente esempio di codice si connette al tuo servizio Hive senza un tunnel.

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

Negli esempi precedenti, sostituisci *hive_user* e *hive_password* con il nome del tuo utente Hive e la password per questo utente.

Quindi, sostituisci *hive_address* e *hive_port* con l'indirizzo NameNode IP e la porta del tuo cluster Hadoop di origine.

Per *hive_remote_address*, puoi utilizzare il valore predefinito 127.0.0.1 o l'indirizzo NameNode IP del servizio Hive di origine.

3. Aggiungi il tuo servizio HDFS di origine al progetto.

Usa il `AddSourceClusterHDFS` comando per connettere il servizio HDFS di origine. Assicurati di fornire valori per i seguenti parametri obbligatori: `userpassword`, `cluster`, `name`, e `port`. Gli altri parametri sono opzionali.

Assicurati che l'utente disponga delle autorizzazioni necessarie per migrare i dati dal servizio HDFS di origine. Per ulteriori informazioni, consulta [Autorizzazioni per l'utilizzo di Hive come fonte](#).

L'esempio di codice seguente crea un tunnel AWS SCT per funzionare con il servizio Apache HDFS. Questo esempio utilizza il cluster `HADOOP_SOURCE` di origine che hai creato in precedenza.

```
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'
```

/

Il codice seguente si connette al tuo servizio Apache HDFS senza un tunnel.

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

/

Negli esempi precedenti, sostituisci *hdfs_user* e *hdfs_password* con il nome dell'utente HDFS e la password di questo utente.

Quindi, sostituisci *hdfs_address* e *hdfs_port* con l'indirizzo NameNode IP e la porta del tuo cluster Hadoop di origine.

Per *hdfs_remote_address*, è possibile utilizzare il valore predefinito 127.0.0.1 o l'indirizzo NameNode IP del servizio Hive di origine.

4. Salva il tuo script CLI. Quindi, aggiungi le informazioni di connessione per il cluster Amazon EMR di destinazione e i comandi di migrazione.

Connessione ad Amazon EMR come destinazione

Puoi connetterti al cluster Amazon EMR di destinazione con l'interfaccia a AWS SCT riga di comando. A tal fine, autorizzi il traffico in entrata e usi SSH. In questo caso, AWS SCT dispone di tutte le autorizzazioni necessarie per lavorare con il tuo cluster Amazon EMR. Per ulteriori informazioni, consulta [Prima della connessione](#) e della [connessione al nodo primario tramite SSH](#) nella Amazon EMR Management Guide.

AWS SCT si connette ad Amazon EMR Hive con l'utente del hadoop cluster. Per connetterti ad Amazon EMR Hive, usa il driver Hive JDBC versione 2.6.2.1002 o superiore. Per ulteriori informazioni, consulta [Scaricamento dei driver di database richiesti](#).

Per connettersi ad Amazon EMR nell'interfaccia a riga di comando AWS SCT

1. Apri il tuo script CLI che include le informazioni di connessione per il tuo cluster Hadoop di origine. Aggiungi le credenziali Amazon EMR di destinazione in questo file.
2. Aggiungi il cluster Amazon EMR di destinazione al progetto.

L'esempio di codice seguente aggiunge il cluster Amazon EMR di destinazione. Questo esempio viene impostato HADOOP_TARGET come nome del cluster di destinazione. Usa questo nome di oggetto per aggiungere i tuoi servizi Hive e HDFS e una cartella bucket Amazon S3 al progetto e creare regole di mappatura.

```
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'
/
```

Nell'esempio precedente, inserisci i nomi AWS delle tue risorse e le informazioni di connessione Amazon EMR. Ciò include l'indirizzo IP del cluster Amazon EMR, la chiave di AWS accesso, la chiave di accesso AWS segreta e il bucket Amazon S3. Se necessario, configura il valore della variabile porta. Quindi, sostituisci *emr_user* ed *emr_password* con il nome del tuo utente Amazon EMR e la password per questo utente. In *path\ name*, inserisci il nome e il percorso del file PEM per il cluster Amazon EMR di destinazione. Per ulteriori informazioni, vedere [Scarica il file PEM per EMR Cluster Access](#).

3. Aggiungi il bucket Amazon S3 di destinazione al progetto.

L'esempio di codice seguente aggiunge il bucket Amazon S3 di destinazione. Questo esempio utilizza il HADOOP_TARGET cluster che hai creato in precedenza.


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

Nell'esempio precedente, inserisci la tua chiave di AWS accesso, la chiave di accesso AWS segreta e il bucket Amazon S3.

4. Aggiungi il servizio Hive di destinazione al progetto.

L'esempio di codice seguente crea un tunnel AWS SCT per funzionare con il servizio Hive di destinazione. Questo esempio utilizza il cluster HADOOP_TARGET di destinazione che hai creato in precedenza.

```
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'
/
```

Nell'esempio precedente, sostituisci *hive_user* e *hive_password* con il nome del tuo utente Hive e la password per questo utente.

Quindi, sostituisci *hive_address* con il valore predefinito 127.0.0.1 o con l'indirizzo NameNode IP del servizio Hive di destinazione. Quindi, sostituisci *hive_port* con la porta del servizio Hive di destinazione.

5. Aggiungi il servizio HDFS di destinazione al progetto.

L'esempio di codice seguente crea un tunnel AWS SCT per funzionare con il servizio Apache HDFS. Questo esempio utilizza il cluster HADOOP_TARGET di destinazione che hai creato in precedenza.

```
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'
```

Nell'esempio precedente, sostituisci *hdfs_user* e *hdfs_password* con il nome dell'utente HDFS e la password di questo utente.

Quindi, sostituisci *hdfs_address* e *hdfs_port* con l'indirizzo IP privato e la porta del NameNode servizio HDFS di destinazione.

6. Salva il tuo script CLI. Quindi, aggiungi le regole di mappatura e i comandi di migrazione. Per ulteriori informazioni, consulta [Migrazione di Apache Hadoop su Amazon EMR](#).

Utilizzo di Apache Oozie come fonte per AWS SCT

È possibile utilizzare l'interfaccia a riga di AWS SCT comando (CLI) per convertire i flussi di lavoro di Apache Oozie in AWS Step Functions. Dopo aver migrato i carichi di lavoro di Apache Hadoop su Amazon EMR, puoi utilizzare un servizio nativo per orchestrare i tuoi lavori. Cloud AWS Per ulteriori informazioni, consulta [Usare Apache Hadoop come fonte](#).

AWS SCT converte i flussi di lavoro di Oozie in AWS Step Functions e li utilizza AWS Lambda per emulare funzionalità che non AWS Step Functions supportano. Inoltre, AWS SCT converte le tue proprietà lavorative di Oozie in AWS Systems Manager

Per convertire i flussi di lavoro di Apache Oozie, assicurati di utilizzare la AWS SCT versione 1.0.671 o superiore. Inoltre, acquisisci familiarità con l'interfaccia a riga di comando di AWS SCT. Per ulteriori informazioni, consulta [AWS SCT Riferimento CLI](#).

Prerequisiti per l'utilizzo di Apache Oozie come sorgente

I seguenti prerequisiti sono necessari per connettersi ad Apache Oozie con la CLI AWS SCT

- Crea un bucket Amazon S3 per archiviare le definizioni delle macchine a stati. È possibile utilizzare queste definizioni per configurare le macchine a stato. Per ulteriori informazioni, consulta [Creazione di un bucket](#) nella Guida per l'utente di Amazon S3.
- Crea un ruolo AWS Identity and Access Management (IAM) con la AmazonS3FullAccess policy. AWS SCT utilizza questo ruolo IAM per accedere al tuo bucket Amazon S3.
- Prendi nota della tua chiave AWS segreta e della tua chiave di accesso AWS segreta. Per ulteriori informazioni sulle chiavi di AWS accesso, consulta [Managing Access Keys](#) nella IAM User Guide.
- Memorizza AWS le tue credenziali e le informazioni sul tuo bucket Amazon S3 nel profilo del AWS servizio nelle impostazioni globali dell'applicazione. Quindi, AWS SCT utilizza questo profilo AWS di servizio per lavorare con AWS le tue risorse. Per ulteriori informazioni, consulta [Archiviazione dei profili AWS di servizio in AWS SCT](#).

Per funzionare con i flussi di lavoro di origine di Apache Oozie, è AWS SCT necessaria la struttura specifica dei file sorgente. Ogni cartella dell'applicazione deve includere il `job.properties` file. Questo file include coppie chiave-valore delle proprietà del lavoro. Inoltre, ognuna delle cartelle dell'applicazione deve includere il `workflow.xml` file. Questo file descrive i nodi di azione e i nodi del flusso di controllo del flusso di lavoro.

Connessione ad Apache Oozie come sorgente

Usa la seguente procedura per connetterti ai tuoi file sorgente di Apache Oozie.

Per connettersi ad Apache Oozie nella CLI AWS SCT

1. Crea un nuovo script AWS SCT CLI o modifica un modello di scenario esistente. Ad esempio, puoi scaricare e modificare il `OozieConversionTemplate.scts` modello. Per ulteriori informazioni, consulta [Ottenere scenari CLI](#).
2. Configurare le impostazioni AWS SCT dell'applicazione.

Il seguente esempio di codice salva le impostazioni dell'applicazione e consente di memorizzare le password nel progetto. Puoi usare queste impostazioni salvate in altri progetti.

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

3. Crea un nuovo AWS SCT progetto.

L'esempio di codice seguente crea il `oozie` progetto nella `c:\sct` cartella.

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

4. Aggiungi la cartella con i tuoi file Apache Oozie di origine al progetto usando il `AddSource` comando. Assicurati di utilizzare il `APACHE_00ZIE` valore per il `vendor` parametro. Inoltre, fornisci valori per i seguenti parametri obbligatori: `name` e `emappingsFolder`.

L'esempio di codice seguente aggiunge Apache Oozie come sorgente nel tuo AWS SCT progetto. Questo esempio crea un oggetto sorgente con il nome `OOZIE`. Usa questo nome di oggetto per aggiungere regole di mappatura. Dopo aver eseguito questo esempio di codice, AWS SCT utilizza la `c:\oozie` cartella per caricare i file sorgente nel progetto.

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

È possibile utilizzare questo esempio e i seguenti in Windows.

5. Connettiti ai tuoi file Apache Oozie di origine usando il `ConnectSource` comando. Usa il nome dell'oggetto di origine che hai definito nel passaggio precedente.

```
ConnectSource
  -name: 'OOZIE'
```

```
-mappingsFolder: 'c:\oozie'  
/
```

6. Salva il tuo script CLI. Quindi, aggiungi le informazioni di connessione per il tuo AWS Step Functions servizio.

Autorizzazioni per l'utilizzo AWS Lambda delle funzioni nel pacchetto di estensione

Per le funzioni di origine che AWS Step Functions non supportano, AWS SCT crea un pacchetto di estensione. Questo pacchetto di estensione include AWS Lambda funzioni che emulano le funzioni di origine.

Per utilizzare questo pacchetto di estensione, crea un ruolo AWS Identity and Access Management (IAM) con le seguenti autorizzazioni.

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

```

Per applicare il pacchetto di estensione, è AWS SCT necessario un ruolo IAM con le seguenti autorizzazioni.

```

{
  "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"
    ]
  }
]
}
```

Connessione a AWS Step Functions come obiettivo

Usa la seguente procedura per connetterti a AWS Step Functions come destinazione.

Per connettersi AWS Step Functions nella AWS SCT CLI

1. Apri il tuo script CLI che include le informazioni di connessione per i tuoi file sorgente di Apache Oozie.
2. Aggiungi le informazioni sul tuo obiettivo di migrazione nel AWS SCT progetto utilizzando il `AddTarget` comando. Assicurati di utilizzare il `STEP_FUNCTIONS` valore per il `vendor` parametro. Inoltre, fornisci valori per i seguenti parametri obbligatori: `name` e `profile`.

L'esempio di codice seguente aggiunge AWS Step Functions come sorgente al tuo AWS SCT progetto. Questo esempio crea un oggetto di destinazione con il nome `AWS_STEP_FUNCTIONS`. Usa questo nome di oggetto quando crei regole di mappatura. Inoltre, questo esempio utilizza un profilo AWS SCT di servizio creato nella fase dei prerequisiti. Assicurati di sostituire *profile_name con il nome* del tuo profilo.

```
AddTarget
  -name: 'AWS_STEP_FUNCTIONS'
```

```
-vendor: 'STEP_FUNCTIONS'  
-profile: 'profile_name'
```

```
/
```

Se non utilizzi il profilo del AWS servizio, assicurati di fornire i valori per i seguenti parametri obbligatori: `accessKeysecretKey`, `awsRegion`, `es3Path`. Usa questi parametri per specificare la tua chiave di accesso AWS AWS segreta Regione AWS, la chiave segreta e il percorso del tuo bucket Amazon S3.

3. Connettiti a AWS Step Functions usando il `ConnectTarget` comando. Usa il nome dell'oggetto di destinazione che hai definito nel passaggio precedente.

Il seguente esempio di codice si connette all'oggetto `AWS_STEP_FUNCTIONS` di destinazione utilizzando il tuo profilo AWS di servizio. Assicurati di sostituire *profile_name con il nome* del tuo profilo.

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

```
/
```

4. Salva il tuo script CLI. Quindi, aggiungi le regole di mappatura e i comandi di migrazione. Per ulteriori informazioni, consulta [Convertire Apache Oozie in AWS Step Functions](#).

Usare il database SQL di Azure come fonte per AWS SCT

Puoi usarlo AWS SCT per convertire schemi, oggetti di codice e codice dell'applicazione dal database SQL di Azure ai seguenti obiettivi:

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

Argomenti

- [Privilegi per il database SQL di Azure come origine](#)
- [Connessione al database SQL di Azure come origine](#)

Privilegi per il database SQL di Azure come origine

I privilegi richiesti per il database SQL di Azure come origine sono i seguenti:

- VIEW DEFINITION
- VIEW DATABASE STATE

Ripeti l'assegnazione per ogni database di cui desideri convertire lo schema.

I privilegi richiesti per i database MySQL e PostgreSQL di destinazione sono descritti nelle sezioni seguenti.

- [Privilegi per MySQL come database di destinazione](#)
- [Privilegi per PostgreSQL come database di destinazione](#)

Connessione al database SQL di Azure come origine

Usa la procedura seguente per connetterti al tuo database di origine del database SQL di Azure con AWS Schema Conversion Tool

Per connettersi a un database di origine del database SQL di Azure

1. NelAWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Azure SQL Database, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCTvisualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per AWSSecret, scegli il nome del segreto.
 2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere[Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine del database SQL di Azure, usa le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Immetti il nome Domain Name Service (DNS) o l'indirizzo IP del server del database di origine.
Database	Inserisci il nome del database a cui connetterti.
User name (Nome utente) e Password (Password)	Inserisci le credenziali del database per connetterti al server del database di origine. AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.
Memorizza la password	AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza dover inserire la password.

5. Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
6. Scegli Connetti per connetterti al tuo database di origine.

Utilizzo di IBM Db2 for z/OS come fonte per AWS SCT

Puoi utilizzarlo AWS SCT per convertire schemi, oggetti di codice e codice applicativo da IBM Db2 for z/OS ai seguenti obiettivi.

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

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

Prerequisiti per Db2 for z/OS come database di origine

La versione del database IBM Db2 for z/OS versione 12 di livello di funzione 100 non supporta la maggior parte delle nuove funzionalità di IBM Db2 for z/OS versione 12. Questa versione del database fornisce il supporto per il fallback alla versione 11 di Db2 e la condivisione dei dati con Db2 versione 11. Per evitare la conversione delle funzionalità non supportate di Db2 versione 11, si consiglia di utilizzare un database IBM Db2 for z/OS di livello 500 o superiore come fonte per. AWS SCT

È possibile utilizzare il seguente esempio di codice per verificare la versione del database IBM Db2 for z/OS di origine.

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

Assicurati che questo codice restituisca una versione DSN12015 o superiore.

È possibile utilizzare il seguente esempio di codice per verificare il valore del registro APPLICATION COMPATIBILITY speciale nel database IBM Db2 for z/OS di origine.

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

Assicurati che questo codice restituisca una versione V12R1M500 o superiore.

Privilegi per Db2 for z/OS come database di origine

I privilegi necessari per connettersi a un database Db2 for z/OS e leggere i cataloghi e le tabelle di sistema sono i seguenti:

- SELEZIONA SU SYSIBM.LOCATIONS
- SELEZIONA SU SYSIBM.SYSCHECKS
- SELEZIONA SU SYSIBM.SYSCOLUMNS
- SELEZIONA SU SYSIBM.SYSDATABASE
- SELEZIONA SU SYSIBM.SYSDATATYPES

- SELEZIONA SYSIBM.SYSDUMMY1
- SELEZIONA SYSIBM.SYSFOREIGNKEYS
- SELEZIONA SU SYSIBM.SYSINDEXES
- SELEZIONA SYSIBM.SYSKEYCLUSE
- SELEZIONA SU SYSIBM.SYSKEYS
- SELEZIONA SU SYSIBM.SYSKEYTARGETS
- SELEZIONA SU SYSIBM.SYSJAROBJECTS
- SELEZIONA SU SYSIBM.SYSPACKAGE
- SELEZIONA SU SYSIBM.SYSPARMS
- SELEZIONA SU SYSIBM.SYSRELS
- SELEZIONA SU SYSIBM.SYSROUTINES
- SELEZIONA SYSIBM.SYSSEQUENCES
- SELEZIONA SYSIBM.SYSSEQUENCESDEP
- SELEZIONA SU SYSIBM.SYSSINONIMI
- SELEZIONA SU SYSIBM.SYSTABCONST
- SELEZIONA SU SYSIBM.SYSTABLES
- SELEZIONA SU SYSIBM.SYSTABLESPACE
- SELEZIONA SU SYSIBM.SYSTRIGGERS
- SELEZIONA SYSIBM.SYSVARIABLES
- SELEZIONA SU SYSIBM.SYSVIEWS

Per convertire le tabelle Db2 for z/OS in tabelle partizionate PostgreSQL, raccogliete statistiche sui tablespaces e sulle tabelle nel database utilizzando l'utilità come illustrato di seguito. RUNSTATS

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

Nell'esempio precedente, sostituire il *YOURDB* segnaposto con il nome del database di origine.

Connessione a Db2 for z/OS come sorgente

Usa la seguente procedura per connetterti al tuo database sorgente Db2 for z/OS con AWS SCT

Per connettersi a un database di origine IBM Db2 for z/OS

1. NelAWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Db2 for z/OS, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCTvisualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per AWSSecret, scegli il nome del segreto.
 2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere[Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine IBM Db2 for z/OS, utilizza le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Inserisci il nome DNS (Domain Name System) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
Location (Ubicazione)	Inserisci il nome univoco della posizione Db2 a cui desideri accedere.
User name (Nome utente) e Password (Password)	Inserisci le credenziali del database per connetterti al server del database di origine.

Parametro	Operazione
	AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.
Use SSL (Usa SSL)	<p>Scegli questa opzione se desideri utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none"> Trust store: la posizione di un trust store contenente certificati. Affinché questa posizione appaia qui, assicurati di aggiungerla nelle Impostazioni globali.
Memorizza la password	AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza dover inserire la password.
Percorso del driver Db2 per z/OS	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

- Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
- Scegli Connetti per connetterti al tuo database di origine.

Privilegi per MySQL come database di destinazione

I privilegi richiesti per MySQL come destinazione sono i seguenti:

- CREA SU * . *
- MODIFICARE SU * . *
- RILASCIARE SU * . *
- INDICE SU * . *
- RIFERIMENTI SU* . *
- SELECT ON *.*
- CREA VISUALIZZAZIONE SU* . *
- SHOW VIEW ON *.*
- GRILLETTO ATTIVATO* . *
- CREA ROUTINE SU * . *
- MODIFICA LA ROUTINE SU * . *
- ESEGUI IL* . *
- SELECT ON mysql.proc
- INSERISCI, AGGIORNA SU AWS_DB2ZOS_EXT. *
- INSERISCI, AGGIORNA, ELIMINA SU AWS_DB2ZOS_EXT_DATA. *
- CREARE TABELLE TEMPORANEE SU AWS_DB2ZOS_EXT_DATA. *

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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';
```

Nell'esempio precedente, sostituisci *user_name* con il nome del tuo utente. Quindi, sostituisci *your_password* con una password sicura.

Per utilizzare Amazon RDS for MySQL come destinazione, imposta il `log_bin_trust_function_creators` parametro su `true` e su `to. character_set_server latin1`. Per configurare questi parametri, create un nuovo gruppo di parametri DB o modificate un gruppo di parametri DB esistente.

Per utilizzare Aurora MySQL come destinazione, imposta il `log_bin_trust_function_creators` parametro su `true` e su `to. character_set_server latin1`. Inoltre, imposta il `lower_case_table_names` parametro su `true`. Per configurare questi parametri, create un nuovo gruppo di parametri DB o modificate un gruppo di parametri DB esistente.

Privilegi per PostgreSQL come database di destinazione

Per utilizzare PostgreSQL come destinazione, è AWS SCT necessario il `CREATE ON DATABASE` privilegio. Assicurati di concedere questo privilegio per ogni database PostgreSQL di destinazione.

Per utilizzare Amazon RDS per PostgreSQL come destinazione, è AWS SCT necessario il privilegio. `rds_superuser`

Per utilizzare i sinonimi pubblici convertiti, modifica il percorso di ricerca predefinito del database in `"$user", public_synonyms, public`.

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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;
```


Nell'esempio precedente, sostituisci *user_name con il nome* del tuo utente. Quindi, sostituisci *db_name con il nome* del tuo database di destinazione. Infine, sostituisci *your_password* con una password sicura.

In PostgreSQL, solo il proprietario dello schema o un `superuser` può eliminare uno schema. Il proprietario può eliminare uno schema e tutti gli oggetti inclusi in questo schema anche se il proprietario dello schema non possiede alcuni dei suoi oggetti.

Quando utilizzi utenti diversi per convertire e applicare schemi diversi al tuo database di destinazione, puoi ricevere un messaggio di errore quando non AWS SCT riesci a eliminare uno schema. Per evitare questo messaggio di errore, usa il `superuser` ruolo.

Impostazioni di conversione da Db2 per z/OS a PostgreSQL

Per modificare le impostazioni di conversione da Db2 for z/OS a PostgreSQL, scegli Impostazioni, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Db2 per z/OS, quindi scegli Db2 per z/OS — PostgreSQL o Db2 per z/OS — Amazon Aurora (compatibile con PostgreSQL). AWS SCT visualizza tutte le impostazioni disponibili per la conversione da IBM Db2 for z/OS a PostgreSQL.

Le impostazioni di conversione da Db2 for z/OS a PostgreSQL AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungi commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per generare nomi univoci per i vincoli nel database di destinazione.

In PostgreSQL, tutti i nomi dei vincoli utilizzati devono essere univoci. AWS SCT può generare nomi univoci per i vincoli nel codice convertito aggiungendo un prefisso con il nome della tabella al nome del vincolo. Per assicurarti che AWS SCT generi nomi univoci per i tuoi vincoli, seleziona Genera nomi univoci per i vincoli.

- Per mantenere la formattazione dei nomi delle colonne, delle espressioni e delle clausole nelle istruzioni DML nel codice convertito.

AWS SCT può mantenere il layout dei nomi delle colonne, delle espressioni e delle clausole nelle istruzioni DML nella posizione e nell'ordine simili a quelli del codice sorgente. A tale scopo, selezionate Sì in Mantieni la formattazione dei nomi delle colonne, delle espressioni e delle clausole nelle istruzioni DML.

- Per escludere le partizioni della tabella dall'ambito di conversione.

AWS SCT può saltare tutte le partizioni di una tabella sorgente durante la conversione. A tale scopo, seleziona Escludi le partizioni della tabella dall'ambito di conversione.

- Per utilizzare il partizionamento automatico per le tabelle partizionate in base alla crescita.

Per la migrazione dei dati, AWS SCT può partizionare automaticamente tutte le tabelle di dimensioni superiori a quelle specificate. Per utilizzare questa opzione, seleziona Applica la partizione di tabelle più grandi di e inserisci la dimensione delle tabelle in gigabyte. Quindi, inserisci il numero di partizioni. AWS SCT considera la dimensione del dispositivo di archiviazione ad accesso diretto (DASD) del database di origine quando si attiva questa opzione.

AWS SCT può determinare automaticamente il numero di partizioni. A tale scopo, seleziona Aumenta proporzionalmente il numero di partizioni e inserisci il numero massimo di partizioni.

- Per restituire set di risultati dinamici come matrice di valori del tipo di dati refcursor.

AWS SCT può convertire le procedure di origine che restituiscono set di risultati dinamici in procedure che hanno una serie di refcursori aperti come parametro di output aggiuntivo. A tale scopo, seleziona Usa una matrice di refcursori per restituire tutti i set di risultati dinamici.

- Per specificare lo standard da utilizzare per la conversione dei valori di data e ora in rappresentazioni di stringhe.

AWS SCT può convertire i valori di data e ora in rappresentazioni di stringhe utilizzando uno dei formati di settore supportati. A tale scopo, seleziona Usa rappresentazioni di stringhe di valori di data o Usa rappresentazioni di stringhe di valori temporali. Quindi, scegli uno dei seguenti standard.

- Organizzazione internazionale per gli standard (ISO)
- Standard europeo IBM (EUR)
- IBM USA Standard (Stati Uniti)
- Standard industriale giapponese Christian Era (JIS)

Utilizzo di IBM Db2 LUW come fonte per AWS SCT

Puoi utilizzarlo AWS SCT per convertire schemi, oggetti di codice nel linguaggio SQL e codice applicativo da IBM Db2 per Linux, Unix e Windows (Db2 LUW) ai seguenti obiettivi.

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

AWS SCT supporta come sorgente Db2 LUW le versioni 9.1, 9.5, 9.7, 10.1, 10.5, 11.1 e 11.5.

Privilegi per Db2 LUW come fonte

I privilegi necessari per connettersi a un database Db2 LUW, per verificare i privilegi disponibili e leggere i metadati dello schema per una fonte sono i seguenti:

- Privilegi necessari per stabilire una connessione:
 - CONNESSIONE AL DATABASE
- Privilegio necessario per l'esecuzione di istruzioni SQL:
 - ESEGUIRE SUL PACCHETTO NULLID.SYSSH200
- Privilegi necessari per ottenere informazioni a livello di istanza:
 - ESEGUE SULLA FUNZIONE SYSPROC.ENV_GET_INST_INFO
 - SELEZIONA SU SYSIBMADM.ENV_INST_INFO
 - SELEZIONA SU SYSIBMADM.ENV_SYS_INFO
- Privilegi necessari a verificare i privilegi concessi tramite ruoli, gruppi e autorità:
 - ESEGUE SULLA FUNZIONE SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID
 - ESEGUE SULLA FUNZIONE SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID
 - ESEGUE SULLA FUNZIONE SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID
 - SELEZIONA SYSIBMADM.PRIVILEGES
- Privilegi necessari per cataloghi di sistema e tabelle:
 - SELEZIONA SU SYSCAT.ATTRIBUTES
 - SELEZIONA SU SYSCAT.CHECKS

- SELEZIONA SU SYSCAT.COLIDENTATTRIBUTES
- SELEZIONA SU SYSCAT.COLUMNS
- SELEZIONA SU SYSCAT.DATAPARTITIONEXPRESSION
- SELEZIONA SU SYSCAT.DATAPARTITIONS
- SELEZIONA SU SYSCAT.DATATYPEDEP
- SELEZIONA SU SYSCAT.DATATYPES
- SELEZIONA SU SYSCAT.HIERARCHIES
- SELEZIONA SYSCAT.INDEXCLUSE
- SELEZIONA SU SYSCAT.INDEXES
- SELEZIONA SU SYSCAT.INDEXPARTITIONS
- SELEZIONA SYSCAT.KEYCLUSE
- SELEZIONA SU SYSCAT.MODULEOBJECTS
- SELEZIONA SU SYSCAT.MODULES
- SELEZIONA SU SYSCAT.NICKNAME
- SELEZIONA SU SYSCAT.PERIODS
- SELEZIONA SU SYSCAT.REFERENCES
- SELEZIONA SU SYSCAT.ROUTINEPARMS
- SELEZIONA SU SYSCAT.ROUTINES
- SELEZIONA SU SYSCAT.ROWFIELDS
- SELEZIONA SU SYSCAT.SCHEMATA
- SELEZIONA SU SYSCAT.SEQUENCES
- SELEZIONA SU SYSCAT.TABCONST
- SELEZIONA SU SYSCAT.TABLES
- SELEZIONA SYSCAT.TRIGGERS
- SELEZIONA SU SYSCAT.VARIABLEDEP
- SELEZIONA SU SYSCAT.VARIABLES
- SELEZIONA SU SYSCAT.VIEWS
- SELEZIONA SYSIBM.SYSDUMMY1

• Per eseguire le istruzioni SQL, l'account utente necessita di un privilegio per l'utilizzo di almeno uno
dei carichi di lavoro abilitati nel database. Se nessuno dei carichi di lavoro è assegnato all'utente,
accertati che il carico di lavoro utente predefinito sia accessibile per l'utente:

- UTILIZZO SUL CARICO DI LAVORO SYSDEFAULTUSERWORKLOAD

Per eseguire le query, devi creare uno spazio tabelle di sistema temporaneo con dimensioni di pagina 8K, 16K e 32K, se non esiste. Per creare spazi tabelle temporanei, esegui gli script seguenti:

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

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_8K
  PAGESIZE 8192
  BUFFERPOOL BP8K;

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

CREATE SYSTEM TEMPORARY TABLESPACE TS_SYS_TEMP_BP16K
  PAGESIZE 16384
  BUFFERPOOL BP16K;

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

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

Connessione a Db2 LUW come sorgente

Utilizza la seguente procedura per connetterti al database di origine Db2 LUW con AWS Schema Conversion Tool.

Per connettersi a un database di origine Db2 LUW

1. NelAWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Db2 LUW, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCTvisualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per AWSSecret, scegli il nome del segreto.
 2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere[Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine IBM Db2 LUW, utilizza le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Inserisci il nome DNS (Domain Name System) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
Database	Immetti il nome del database Db2 LUW.
User name (Nome utente) e Password (Password)	Inserisci le credenziali del database per connetterti al server del database di origine. AWS SCTutilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto

Parametro	Operazione
	AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.
Use SSL (Usa SSL)	<p>Scegli questa opzione se desideri utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none"> Trust store: la posizione di un trust store contenente certificati. Affinché questa posizione appaia qui, assicurati di aggiungerla nelle Impostazioni globali.
Memorizza la password	AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza dover inserire la password.
Percorso del driver Db2 LUW	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

- Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
- Scegli Connetti per connetterti al tuo database di origine.

Conversione di Db2 LUW in Amazon RDS per PostgreSQL o versione compatibile con Amazon Aurora PostgreSQL

Quando esegui la migrazione IBM Db2 LUW a PostgreSQL, AWS SCT può convertire varie istruzioni trigger utilizzate con Db2 LUW. Queste istruzioni trigger includono quanto segue:

- **Eventi di attivazione:** gli eventi di attivazione INSERT, DELETE e UPDATE specificano che l'azione attivata viene eseguita ogni volta che l'evento viene applicato alla tabella dell'oggetto o alla vista dell'argomento. È possibile specificare qualsiasi combinazione degli eventi INSERT, DELETE e UPDATE, ma è possibile specificare ogni evento una sola volta. AWS SCT supporta eventi trigger singoli e multipli. Per gli eventi, PostgreSQL ha praticamente la stessa funzionalità.
- **Evento DI COLONNA:** è possibile specificare il nome di una colonna da una tabella di base. Il trigger viene attivato solo tramite l'aggiornamento di una colonna identificata nell'elenco dei nomi delle colonne. PostgreSQL ha la stessa funzionalità.
- **Trigger dell'istruzione:** specificano che l'azione attivata viene applicata una sola volta per l'intera istruzione. Non è possibile specificare questo tipo di granularità per attivare un trigger BEFORE o un trigger INSTEAD OF. Se specificato, viene attivato un trigger UPDATE o DELETE, anche se non sono coinvolte righe. PostgreSQL, inoltre, dispone di questa funzionalità e la dichiarazione del trigger per i trigger dell'istruzione è identica per PostgreSQL e Db2 LUW.
- **Clausole di riferimento:** specificano i nomi di correlazione per le variabili di transizione e i nomi delle tabelle per le tabelle di transizione. I nomi di correlazione identificano una determinata riga nella serie di righe interessate dall'attivazione SQL. I nomi della tabella identificano il set completo di righe interessate. Ogni riga influenzata da un'operazione di attivazione SQL è disponibile per l'azione attivata qualificando le colonne con nomi di correlazione specifici. PostgreSQL non supporta questa funzionalità e usa solo un nome di correlazione NEW oppure OLD.
- **INVECE DEI trigger:** li AWS SCT supporta.

Conversione di tabelle partizionate Db2 LUW in tabelle partizionate PostgreSQL versione 10

AWS SCT è in grado di convertire tabelle Db2 LUW in tabelle partizionate in PostgreSQL 10. Durante la conversione di una tabella partizionata Db2 LUW in PostgreSQL esistono diverse restrizioni:

- Puoi creare una tabella partizionata con una colonna nullable in Db2 LUW e specificare una partizione per archiviare valori NULL. Tuttavia, PostgreSQL non supporta valori NULL per il partizionamento RANGE.
- Db2 LUW può utilizzare una clausola INCLUSIVE o EXCLUSIVE per impostare valori di delimitazione dell'intervallo. PostgreSQL supporta solo la clausola INCLUSIVE per una delimitazione di inizio e una clausola EXCLUSIVE per una delimitazione di fine. Il nome di partizione convertito è nel formato <nome_tabella_originale>_<nome_partizione_originale>.

- È possibile creare chiavi primarie o univoche per tabelle partizionate in Db2 LUW. PostgreSQL richiede la creazione di una chiave primaria o univoca per ogni partizione direttamente. Le limitazioni della chiave primaria o univoca devono essere rimosse dalla tabella padre. Il nome chiave convertito è nel formato <nome_chiave_originale>_<nome_partizione_originale>.
- Puoi creare un vincolo di chiave esterna da e verso una tabella partizionata in Db2 LUW. Tuttavia, PostgreSQL non supporta i riferimenti delle chiavi esterne nelle tabelle partizionate. Inoltre, PostgreSQL non supporta i riferimenti delle chiavi esterne da una tabella partizionata a un'altra tabella.
- Puoi creare un indice su una tabella partizionata in Db2 LUW. Tuttavia, PostgreSQL richiede la creazione di un indice per ogni partizione direttamente. Gli indici devono essere rimossi dalla tabella padre. Il nome indice convertito è nel formato <nome_indice_originale>_<nome_partizione_originale>.
- È necessario definire trigger di riga su singole partizioni, non sulla tabella partizionata. I trigger devono essere rimossi dalla tabella padre. Il nome trigger convertito è nel formato <nome_trigger_originale>_<nome_partizione_originale>.

Privilegi per PostgreSQL come obiettivo

Per utilizzare PostgreSQL come destinazione, è AWS SCT necessario il CREATE ON DATABASE privilegio. Assicurati di concedere questo privilegio per ogni database PostgreSQL di destinazione.

Per utilizzare i sinonimi pubblici convertiti, modifica il percorso di ricerca predefinito del database in"\$user", public_synonyms, public.

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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;
```

Nell'esempio precedente, sostituisci *user_name con il nome* del tuo utente. Quindi, sostituisci *db_name con il nome* del tuo database di destinazione. Infine, sostituisci *your_password* con una password sicura.

In PostgreSQL, solo il proprietario dello schema o un superuser può eliminare uno schema. Il proprietario può eliminare uno schema e tutti gli oggetti inclusi in questo schema anche se il proprietario dello schema non possiede alcuni dei suoi oggetti.

Quando utilizzi utenti diversi per convertire e applicare schemi diversi al tuo database di destinazione, puoi ricevere un messaggio di errore quando non AWS SCT riesci a eliminare uno schema. Per evitare questo messaggio di errore, usa il `superuser` ruolo.

Conversione da Db2 LUW ad Amazon RDS per MySQL o Amazon Aurora MySQL

Quando converti un database IBM Db2 LUW in RDS per MySQL o Amazon Aurora MySQL, tieni presente quanto segue.

Privilegi per MySQL come obiettivo

I privilegi richiesti per MySQL come destinazione sono i seguenti:

- CREA SU * . *
- MODIFICARE SU * . *
- RILASCIARE SU * . *
- INDICE SU * . *
- RIFERIMENTI SU* . *
- SELECT ON *.*
- CREA VISUALIZZAZIONE SU* . *
- SHOW VIEW ON *.*
- GRILLETTO ATTIVATO* . *
- CREA ROUTINE SU * . *
- MODIFICA LA ROUTINE SU * . *
- ESEGUI IL* . *
- SELECT ON mysql.proc
- INSERISCI, AGGIORNA SU AWS_DB2_EXT. *
- INSERISCI, AGGIORNA, ELIMINA SU AWS_DB2_EXT_DATA. *
- CREARE TABELLE TEMPORANEE SU AWS_DB2_EXT_DATA. *

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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';
```

Nell'esempio precedente, sostituisci *user_name* con il nome del tuo utente. Quindi, sostituisci *your_password* con una password sicura.

Per utilizzare Amazon RDS for MySQL o Aurora MySQL come destinazione, imposta il parametro su `lower_case_table_names 1`. Questo valore indica che il server MySQL gestisce gli identificatori di nomi di oggetti come tabelle, indici, trigger e database senza distinzione tra maiuscole e minuscole. Se hai attivato la registrazione binaria nell'istanza di destinazione, imposta il `log_bin_trust_function_creators` parametro su 1. In questo caso, non è necessario utilizzare `NO SQL le READS SQL DATA` o `DETERMINISTIC` le caratteristiche per creare funzioni memorizzate. Per configurare questi parametri, create un nuovo gruppo di parametri DB o modificate un gruppo di parametri DB esistente.

Usare MySQL come fonte per AWS SCT

È possibile utilizzarli AWS SCT per convertire schemi, oggetti di codice del database e codice dell'applicazione da MySQL ai seguenti obiettivi:

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

Per ulteriori informazioni, consulta le sezioni seguenti:

Argomenti

- [Privilegi per MySQL come database di origine](#)
- [Connessione a MySQL come sorgente](#)
- [Privilegi per PostgreSQL come database di destinazione](#)

Privilegi per MySQL come database di origine

I privilegi richiesti per MySQL come sorgente sono i seguenti:

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

Connessione a MySQL come sorgente

Utilizza la seguente procedura per connetterti al database di origine MySQL con AWS Schema Conversion Tool.

Per connettersi al database di origine MySQL

1. NelAWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli MySQL, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. Per Nome di connessione, immettere un nome per il database. AWS SCTvisualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del databaseAWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, utilizza le seguenti istruzioni:
 1. Per AWS Secret, scegli il nome del segreto.
 2. Scegli Popola per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere[Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine MySQL, utilizzare le seguenti istruzioni:

Parametro	Operazione
Use SSL (Usa SSL)	<p>Scegli questa opzione per utilizzare il protocollo SSL (Secure Sockets Layer) per connetterti al database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none">• Richiedi SSL: scegli questa opzione per connetterti al server solo tramite SSL. <p>Se scegli Require SSL (Richiedi SSL), significa che se il server non supporta SSL, non puoi connetterti al server. Se non scegli Require SSL (Richiedi SSL) e il server non supporta SSL, puoi ancora connetterti al server senza utilizzare SSL. Per ulteriori informazioni, consulta Configurazione di MySQL per l'utilizzo di connessioni sicure.</p> <ul style="list-style-type: none">• Verifica certificato del server: seleziona questa opzione per verificare il certificato del server utilizzando un trust store.• Trust store: l'ubicazione di un trust store contenente certificati.
Memorizza la password	<p>AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Abilitando questa opzione potrai archiviare la password del database e connetterti in modo rapido al database senza dover inserire la password.</p>
MySQL percorso del conducente	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

5. Scegli Connessione di prova per verificare che sia inAWS SCT grado di connettersi al database di origine.
6. Scegli Connect per connetterti al tuo database di origine.

Privilegi per PostgreSQL come database di destinazione

Per utilizzare PostgreSQL come destinazione, èAWS SCT necessario ilCREATE ON DATABASE privilegio. Assicurati di concedere questo privilegio per ogni database PostgreSQL di destinazione.

Per utilizzare i sinonimi pubblici convertiti, modifica il percorso di ricerca predefinito del database in"\$user", public_synonyms, public.

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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;
```

Nell'esempio precedente, sostituisci *user_name* con il nome del tuo utente. Quindi, sostituisci *db_name* con il nome del tuo database di destinazione. Infine, sostituisci *your_password* con una password sicura.

In PostgreSQL, solo il proprietario dello schema o unsuperuser può eliminare uno schema. Il proprietario può eliminare uno schema e tutti gli oggetti che questo schema include anche se il proprietario dello schema non possiede alcuni dei suoi oggetti.

Quando si utilizzano utenti diversi per convertire e applicare schemi diversi al database di destinazione, è possibile ricevere un messaggio di errore quando non èAWS SCT possibile eliminare uno schema. Per evitare questo messaggio di errore, usa ilsuperuser ruolo.

Utilizzo del database Oracle come fonte per AWS SCT

È possibile utilizzare AWS SCT per convertire schemi, oggetti di codice del database e codice dell'applicazione da Oracle Database ai seguenti obiettivi:

- Amazon RDS for MySQL

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

Quando l'origine è un database Oracle, i commenti possono essere convertiti nel formato appropriato, ad esempio, in un database PostgreSQL. AWS SCT può convertire commenti su tabelle, viste e colonne. I commenti possono includere apostrofi; AWS SCT raddoppia gli apostrofi durante la conversione delle istruzioni SQL, proprio come avviene per le stringhe letterali.

Per ulteriori informazioni, consultare gli argomenti seguenti.

Argomenti

- [Privilegi per Oracle come fonte](#)
- [Connessione a Oracle come fonte](#)
- [Conversione da Oracle ad Amazon RDS per PostgreSQL o Amazon Aurora PostgreSQL](#)
- [Conversione da Oracle ad Amazon RDS per MySQL o Amazon Aurora MySQL](#)
- [Conversione da Oracle ad Amazon RDS per Oracle](#)

Privilegi per Oracle come fonte

I privilegi richiesti per Oracle come fonte sono i seguenti:

- CONNECT
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY
- SELEZIONA SYS.ARGUMENT\$

Connessione a Oracle come fonte

Utilizza la seguente procedura per connetterti al database di origine Oracle con AWS Schema Conversion Tool.

Per connetterti a un database di origine Oracle

1. NelAWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Oracle, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCTvisualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per AWS Secret, scegli il nome del segreto.
 2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine Oracle, utilizzare le seguenti istruzioni:

Parametro	Operazione
Type (Tipo)	<p>Scegli il tipo di connessione al database. A seconda del tipo, fornisci le seguenti informazioni aggiuntive:</p> <ul style="list-style-type: none"> • SID <ul style="list-style-type: none"> • Nome server: il nome DNS (Domain Name System) o l'indirizzo IP del server del database di origine. • Server port (Porta del server): la porta utilizzata per la connessione al server del database di origine. • Oracle SID (SID Oracle): l'ID del sistema Oracle (SID). Per individuare il SID Oracle, invia la seguenti query al database Oracle: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre> • Nome del servizio

Parametro	Operazione
	<ul style="list-style-type: none">• Server name (Nome del server): il nome DNS o l'indirizzo IP del server del database di origine. <p>È possibile connettersi al database Oracle di origine utilizzando un protocollo di indirizzo IPv6. A tale scopo, assicurati di utilizzare le parentesi quadre per inserire l'indirizzo IP, come mostrato nell'esempio seguente.</p> <div data-bbox="719 554 1507 632" 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">• Server port (Porta del server): la porta utilizzata per la connessione al server del database di origine.• Nome servizio: il nome del servizio Oracle a cui connettersi.• Alias TNS<ul style="list-style-type: none">• TNS file path (Percorso del file TNS): il percorso al file che contiene le informazioni sulla connessione al nome del Transparent Network Substrate (TNS).<p>Dopo aver scelto il file TNS, AWS SCT aggiunge tutte le connessioni al database Oracle dal file all'elenco degli alias TNS.</p><p>Scegli questa opzione per connetterti a Oracle Real Application Clusters (RAC).</p>• Alias TNS: l'alias TNS di questo file da utilizzare per connettersi al database di origine. • Identificatore di connessione TNS<ul style="list-style-type: none">• Identificatore di connessione TNS: l'identificatore per le informazioni di connessione TNS registrate.

Parametro	Operazione
User name (Nome utente) e Password (Password)	<p>Inserisci le credenziali del database per connetterti al server del database di origine.</p> <p>La prima volta che ci si connette al database Oracle, si immette il percorso del file Oracle Driver (ojdbc8.jar). Puoi scaricare il file all'indirizzo http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html. Assicurati di registrarti sul sito Web gratuito di Oracle Technical Network per completare il download. AWS SCT utilizza il driver selezionato per eventuali connessioni future al database Oracle. Il percorso del driver può essere modificato utilizzando la scheda Driver in Impostazioni globali.</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p>
Use SSL (Usa SSL)	<p>Scegli questa opzione per utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none">• Autenticazione SSL: seleziona questa opzione per utilizzare l'autenticazione SSL tramite certificato. Configura l'archivio attendibile e l'archivio chiavi in Impostazioni, Impostazioni globali, Sicurezza.• Trust store: l'archivio fiduciario da utilizzare.• Archivio chiavi: l'archivio chiavi da utilizzare.

Parametro	Operazione
Memorizza la password	AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Scegliete questa opzione per memorizzare la password del database e connettervi rapidamente al database senza dover inserire la password.
Percorso del driver Oracle	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se memorizzate il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo di connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

- Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
- Scegli Connetti per connetterti al tuo database di origine.

Conversione da Oracle ad Amazon RDS per PostgreSQL o Amazon Aurora PostgreSQL

Quando converti un database Oracle in RDS per PostgreSQL o Amazon Aurora PostgreSQL, tieni presente quanto segue.

Argomenti

- [Privilegi per PostgreSQL come database di destinazione](#)
- [Impostazioni di conversione da Oracle a PostgreSQL](#)
- [Conversione di sequenze Oracle](#)
- [Conversione di ROWID Oracle](#)
- [Conversione di Oracle Dynamic SQL](#)
- [Conversione di partizioni Oracle](#)

Durante la conversione di oggetti di sistema Oracle in PostgreSQL, AWS SCT esegue le conversioni come illustrato nella tabella seguente.

Oggetto di sistema Oracle	Descrizione	Oggetto PostgreSQL convertito
V\$VERSION	Visualizza i numeri di versione dei componenti principali della libreria nel database Oracle	aws_oracle_ext.v\$version
V\$INSTANCE	Vista che mostra lo stato dell'istanza corrente.	aws_oracle_ext.v\$instance

Puoi usarlo AWS SCT per convertire i file Oracle SQL*Plus in psql, che è un front-end per PostgreSQL basato su terminale. Per ulteriori informazioni, consulta [Conversione del codice SQL dell'applicazione utilizzando AWS SCT](#).

Privilegi per PostgreSQL come database di destinazione

Per utilizzare PostgreSQL come destinazione, è AWS SCT necessario il CREATE ON DATABASE privilegio. Assicurati di concedere questo privilegio per ogni database PostgreSQL di destinazione.

Per utilizzare i sinonimi pubblici convertiti, modifica il percorso di ricerca predefinito del database in "\$user", public_synonyms, public.

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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;
```

Nell'esempio precedente, sostituisci *user_name* con il nome del tuo utente. Quindi, sostituisci *db_name* con il nome del tuo database di destinazione. Infine, sostituisci *your_password* con una password sicura.

Per utilizzare Amazon RDS per PostgreSQL come destinazione, è AWS SCT necessario il privilegio `rds_superuser`

In PostgreSQL, solo il proprietario dello schema o un `superuser` può eliminare uno schema. Il proprietario può eliminare uno schema e tutti gli oggetti inclusi in questo schema anche se il proprietario dello schema non possiede alcuni dei suoi oggetti.

Quando utilizzi utenti diversi per convertire e applicare schemi diversi al tuo database di destinazione, puoi ricevere un messaggio di errore quando non AWS SCT riesci a eliminare uno schema. Per evitare questo messaggio di errore, usa il `superuser` ruolo.

Impostazioni di conversione da Oracle a PostgreSQL

Per modificare le impostazioni di conversione da Oracle a PostgreSQL, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Oracle, quindi scegli Oracle — PostgreSQL. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da Oracle a PostgreSQL.

Le impostazioni di conversione da Oracle a PostgreSQL AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungere commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per consentire di AWS SCT convertire le viste materializzate di Oracle in tabelle o viste materializzate su PostgreSQL. Per la conversione delle viste materializzate come, scegli come convertire le viste materializzate di origine.
- Per lavorare con il codice Oracle sorgente quando include le `TO_NUMBER` funzioni `TO_CHAR` `TO_DATE`, e con parametri non supportati da PostgreSQL. Per impostazione predefinita, AWS SCT emula l'utilizzo di questi parametri nel codice convertito.

Quando il codice Oracle sorgente include solo parametri supportati da PostgreSQL, puoi utilizzare PostgreSQL `TO_CHAR` e funzioni native. `TO_DATE` `TO_NUMBER` In questo caso, il codice convertito funziona più velocemente. Per includere solo questi parametri, selezionate i seguenti valori:

- La funzione `TO_CHAR ()` non utilizza stringhe di formattazione specifiche di Oracle
- La funzione `TO_DATE ()` non utilizza stringhe di formattazione specifiche di Oracle

- La funzione `TO_NUMBER ()` non utilizza stringhe di formattazione specifiche di Oracle
- Per risolvere i casi in cui il database Oracle di origine memorizza solo valori interi nelle colonne chiave primarie o esterne del tipo di `NUMBER` dati, è AWS SCT possibile convertire queste colonne nel tipo di `BIGINT` dati. Questo approccio migliora le prestazioni del codice convertito. Per adottare questo approccio, seleziona `Converti` le colonne di chiave primaria/esterna `NUMBER` in colonne `BIGINT`. Assicurati che la tua fonte non includa valori a virgola mobile in queste colonne per evitare la perdita di dati.
- Per saltare i trigger e i vincoli disattivati nel codice sorgente. A tale scopo, scegli `Ignora` i trigger e i vincoli disabilitati.
- Da utilizzare AWS SCT per convertire variabili di stringa chiamate SQL dinamico. Il codice del database può modificare i valori di queste variabili di stringa. Per assicurarti che converta AWS SCT sempre il valore più recente di questa variabile di stringa, seleziona `Converti` il codice SQL dinamico creato nelle routine chiamate.
- Per risolvere il problema, PostgreSQL versione 10 e precedenti non supporta le procedure. Se tu o i tuoi utenti non avete familiarità con l'uso delle procedure in PostgreSQL, AWS SCT potete convertire le procedure Oracle in funzioni PostgreSQL. A tale scopo, seleziona `Converti` procedure in funzioni.
- Per visualizzare informazioni aggiuntive sulle azioni eseguite. A tale scopo, puoi aggiungere funzioni specifiche al pacchetto di estensione selezionando `Aggiungi` un blocco di segnalazione delle eccezioni per problemi di migrazione con i livelli di gravità successivi. Quindi scegli i livelli di gravità per aumentare le eccezioni definite dall'utente.
- Per lavorare con un database Oracle di origine che potrebbe includere vincoli con i nomi generati automaticamente. Se il codice sorgente utilizza questi nomi, assicuratevi di selezionare `Converti` i nomi dei vincoli generati dal sistema utilizzando i nomi originali di origine. Se il codice sorgente utilizza questi vincoli ma non ne usa i nomi, deseleziona questa opzione per aumentare la velocità di conversione.
- Per stabilire se il database e le applicazioni vengono eseguiti in fusi orari diversi. Per impostazione predefinita, AWS SCT emula i fusi orari nel codice convertito. Tuttavia, questa emulazione non è necessaria quando il database e le applicazioni utilizzano lo stesso fuso orario. In questo caso, seleziona `Il fuso orario sul lato client corrisponde al fuso orario sul server`.
- Per stabilire se i database di origine e di destinazione vengono eseguiti in fusi orari diversi. In tal caso, la funzione che emula la funzione Oracle `SYSDATE` integrata restituisce valori diversi rispetto alla funzione di origine. Per assicurarti che le funzioni di origine e destinazione restituiscano gli stessi valori, scegli `Imposta il fuso orario predefinito per l'emulazione SYSDATE`.

- Per utilizzare le funzioni dell'estensione `orafce` nel codice convertito. A tale scopo, in Usa l'implementazione di `orafce`, seleziona le funzioni da utilizzare. Per ulteriori informazioni su `orafce`, vedere [orafce](#) su. GitHub

Conversione di sequenze Oracle

AWS SCT converte le sequenze da Oracle a PostgreSQL. Se utilizzate le sequenze per mantenere i vincoli di integrità, assicuratevi che i nuovi valori di una sequenza migrata non si sovrappongano ai valori esistenti.

Per compilare le sequenze convertite con l'ultimo valore del database di origine

1. Apri il tuo AWS SCT progetto con Oracle come fonte.
2. Scegli Impostazioni, quindi scegli Impostazioni di conversione.
3. Dall'elenco superiore, scegli Oracle, quindi scegli Oracle — PostgreSQL. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da Oracle a PostgreSQL.
4. Scegli Compila sequenze convertite con l'ultimo valore generato sul lato sorgente.
5. Scegliete OK per salvare le impostazioni e chiudere la finestra di dialogo Impostazioni di conversione.

Conversione di ROWID Oracle

In un database Oracle, la pseudocolonna ROWID contiene l'indirizzo della riga di tabella. La pseudocolonna ROWID è unica per Oracle, quindi AWS SCT converte la pseudocolonna ROWID in una colonna di dati su PostgreSQL. Utilizzando questa conversione, puoi conservare le informazioni ROWID.

Quando converti la pseudocolonna ROWID, AWS SCT puoi creare una colonna di dati con il tipo di dati `bigint`. Se non esiste alcuna chiave primaria, AWS SCT imposta la colonna ROWID come chiave primaria. Se esiste una chiave primaria, AWS SCT imposta la colonna ROWID con un vincolo univoco.

Se il codice del database di origine include operazioni con ROWID, che non puoi eseguire utilizzando un tipo di dati numerico, AWS SCT puoi creare una colonna di dati con il `character varying` tipo di dati.

Per creare una colonna di dati per ROWID Oracle per un progetto

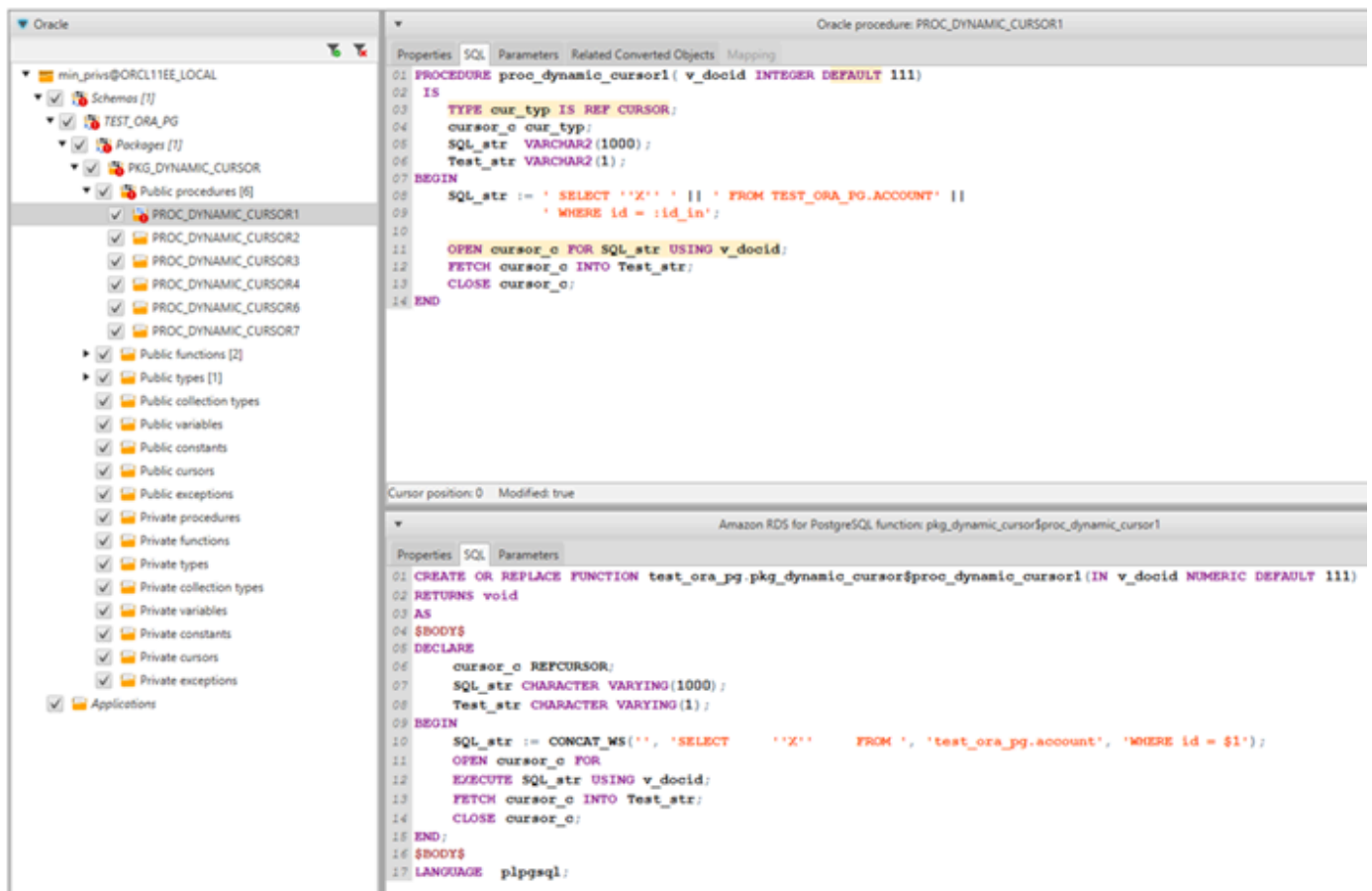
1. Apri il tuo AWS SCT progetto con Oracle come fonte.
2. Scegli Impostazioni, quindi scegli Impostazioni di conversione.
3. Dall'elenco superiore, scegli Oracle, quindi scegli Oracle — PostgreSQL. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da Oracle a PostgreSQL.
4. Per Genera ID di riga, esegui una delle seguenti operazioni:
 - Scegli Genera come identità per creare una colonna di dati numerici.
 - Scegli Genera come tipo di dominio di caratteri per creare una colonna di dati sui caratteri.
5. Scegliete OK per salvare le impostazioni e chiudere la finestra di dialogo Impostazioni di conversione.

Conversione di Oracle Dynamic SQL

Oracle fornisce due modi per implementare l'SQL dinamico: utilizzando un'istruzione EXECUTE IMMEDIATE o richiamando le procedure nel pacchetto DBMS_SQL. Se il database Oracle di origine include oggetti con SQL dinamico, utilizzalo AWS SCT per convertire le istruzioni SQL dinamiche di Oracle in PostgreSQL.

Per convertire Oracle Dynamic SQL in PostgreSQL

1. Apri il tuo AWS SCT progetto con Oracle come fonte.
2. Scegliete un oggetto di database che utilizza SQL dinamico nella vista ad albero dei sorgenti di Oracle.
3. Apri il menu contestuale (clic con il pulsante destro del mouse) per l'oggetto, scegli Converti schema e accetta di sostituire gli oggetti se esistono. La schermata seguente mostra la procedura convertita sotto la procedura Oracle con SQL dinamico.



Conversione di partizioni Oracle

AWS SCT attualmente supporta i seguenti metodi di partizionamento:

- Intervallo
- Elenco
- Intervallo a più colonne
- Hash
- Composito (list-list, range-list, list-range, list-hash, range-hash, hash-hash)

Conversione da Oracle ad Amazon RDS per MySQL o Amazon Aurora MySQL

Per emulare le funzioni del database Oracle nel codice MySQL convertito, usa il pacchetto di estensione da Oracle a MySQL in AWS SCT. Per ulteriori informazioni sui pacchetti di estensione, vedere [Utilizzo dei pacchetti di AWS SCT estensione](#).

Argomenti

- [Privilegi per MySQL come database di destinazione](#)
- [Impostazioni di conversione da Oracle a MySQL](#)
- [Considerazioni sulla migrazione](#)
- [Conversione dell'istruzione WITH in Oracle in RDS per MySQL o Amazon Aurora MySQL](#)

Privilegi per MySQL come database di destinazione

I privilegi richiesti per MySQL come destinazione sono i seguenti:

- CREA SU * . *
- MODIFICARE * . *
- RILASCIARE SU * . *
- INDICE SU * . *
- RIFERIMENTI SU* . *
- SELECT ON *.*
- CREA VISUALIZZAZIONE SU* . *
- SHOW VIEW ON *.*
- GRILLETTO ATTIVATO* . *
- CREA ROUTINE SU * . *
- MODIFICA LA ROUTINE SU * . *
- ESEGUI IL* . *
- CREARE TABELLE TEMPORANEE SU* . *
- AWS_LAMBDA_ACCESS
- INSERISCI, AGGIORNA SU AWS_ORACLE_EXT. *
- INSERISCI, AGGIORNA, ELIMINA SU AWS_ORACLE_EXT_DATA. *

Se utilizzi un database MySQL versione 5.7 o precedente come destinazione, concedi l'autorizzazione `INVOKE LAMBDA *.*` anziché `AWS_LAMBDA_ACCESS`. Per i database MySQL versione 8.0 e successive, concedi l'autorizzazione `AWS_LAMBDA_ACCESS`.

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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';
```

Nell'esempio precedente, sostituisci *user_name* con il nome del tuo utente. Quindi, sostituisci *your_password* con una password sicura.

Se usi un database MySQL versione 5.7 o precedente come destinazione, usa `GRANT INVOKE LAMBDA ON *.* TO 'user_name'` invece di `GRANT AWS_LAMBDA_ACCESS TO 'user_name'`

Per utilizzare Amazon RDS for MySQL o Aurora MySQL come destinazione, imposta il parametro su `lower_case_table_names 1`. Questo valore indica che il server MySQL gestisce gli identificatori di nomi di oggetti come tabelle, indici, trigger e database senza distinzione tra maiuscole e minuscole. Se hai attivato la registrazione binaria nell'istanza di destinazione, imposta il `log_bin_trust_function_creators` parametro su 1. In questo caso, non è necessario utilizzare `NO SQL le READS SQL DATA` o `DETERMINISTIC` le caratteristiche per creare funzioni memorizzate. Per configurare questi parametri, create un nuovo gruppo di parametri DB o modificate un gruppo di parametri DB esistente.

Impostazioni di conversione da Oracle a MySQL

Per modificare le impostazioni di conversione da Oracle a MySQL, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Oracle, quindi scegli Oracle — MySQL. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da Oracle a MySQL.

Le impostazioni di conversione da Oracle a MySQL AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungere commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per risolvere questo problema, il database Oracle di origine può utilizzare la ROWID pseudocolonna, ma MySQL non supporta funzionalità simili. AWS SCT può emulare la ROWID pseudocolonna nel codice convertito. Per farlo, scegli Genera come identità per Genera ID di riga?

Se il tuo codice Oracle sorgente non utilizza la **ROWID** pseudocolonna, scegli Non generare per Genera ID di riga? In questo caso, il codice convertito funziona più velocemente.

- Per lavorare con il codice Oracle sorgente quando include le TO_NUMBER funzioni TO_CHAR TO_DATE, e con parametri non supportati da MySQL. Per impostazione predefinita, AWS SCT emula l'utilizzo di questi parametri nel codice convertito.

Quando il codice Oracle sorgente include solo parametri supportati da PostgreSQL, puoi utilizzare MySQL e funzioni native TO_CHAR. TO_DATE TO_NUMBER In questo caso, il codice convertito funziona più velocemente. Per includere solo questi parametri, selezionate i seguenti valori:

- La funzione TO_CHAR () non utilizza stringhe di formattazione specifiche di Oracle
- La funzione TO_DATE () non utilizza stringhe di formattazione specifiche di Oracle
- La funzione TO_NUMBER () non utilizza stringhe di formattazione specifiche di Oracle
- Per stabilire se il database e le applicazioni vengono eseguiti in fusi orari diversi. Per impostazione predefinita, AWS SCT emula i fusi orari nel codice convertito. Tuttavia, questa emulazione non è necessaria quando il database e le applicazioni utilizzano lo stesso fuso orario. In questo caso, seleziona Il fuso orario sul lato client corrisponde al fuso orario sul server.

Considerazioni sulla migrazione

Quando converti Oracle in RDS per MySQL o Aurora MySQL, per modificare l'ordine in cui vengono eseguite le istruzioni, puoi utilizzare un'istruzione e un'etichetta. GOTO Qualsiasi istruzione PL/SQL che segue un'GOTOistruzione viene ignorata e l'elaborazione continua sull'etichetta. È possibile utilizzare GOTO istruzioni ed etichette ovunque all'interno di una procedura, di un batch o di un blocco di istruzioni. Puoi anche consultare le dichiarazioni GOTO successive.

MySQL non utilizza GOTO istruzioni. Quando AWS SCT converte il codice che contiene un'GOTOistruzione, converte l'istruzione per utilizzare un'istruzione BEGIN...END orLOOP...END LOOP.

Puoi trovare esempi di come AWS SCT converte GOTO le istruzioni nella tabella seguente.

Istruzione Oracle	Istruzione 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; statement4; </pre>	<pre> BEGIN statement1; label1: LOOP statement2; ITERATE label1; LEAVE label1; END LOOP; </pre>

Istruzione Oracle	Istruzione MySQL
<pre> END </pre>	<pre> statement3; statement4; END </pre>
<pre> BEGIN statement1; label1: statement2; statement3; statement4; END </pre>	<pre> BEGIN statement1; label1: BEGIN statement2; statement3; statement4; END; END </pre>

Conversione dell'istruzione WITH in Oracle in RDS per MySQL o Amazon Aurora MySQL

È possibile utilizzare la clausola WITH (subquery_factoring) in Oracle per assegnare un nome (query_name) a un blocco di sottoquery. È quindi possibile fare riferimento al blocco di sottoquery in più posizioni nella query specificando il query_name. Se un blocco di subquery non contiene collegamenti o parametri (locale, procedura, funzione, pacchetto), AWS SCT converte la clausola in una vista o in una tabella temporanea.

Il vantaggio di convertire la clausola in una tabella temporanea è che i riferimenti ripetuti alla sottoquery potrebbero essere più efficienti. La maggiore efficienza è dovuta al fatto che i dati vengono facilmente recuperati dalla tabella temporanea piuttosto che essere richiesti da ogni riferimento. Puoi emularlo utilizzando viste aggiuntive o una tabella temporanea. Il nome della visualizzazione utilizza il formato <procedure_name>\${<subselect_alias>.

Puoi trovare degli esempi nella tabella seguente.

Istruzione Oracle	Istruzione 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_empl(id, name, surname, lastname, state, dept_id) AS (SELECT id, name, surname, lastname, state, dept_id FROM test_ora_ pg.dept_employees WHERE state = p_state AND dept_id = l_dept_id) SELECT id,state FROM dept_empl ORDER BY id) LOOP NULL; END LOOP; </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_empl ORDER BY ID; DECLARE CONTINUE HANDLER FOR NOT FOUND SET done := TRUE; OPEN cur; read_label: LOOP FETCH cur INTO var\$id, var \$state; IF done THEN LEAVE read_label; END IF; BEGIN END; END LOOP; CLOSE cur; </pre>

Istruzione Oracle	Istruzione MySQL
	END;

Istruzione 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;

```

Istruzione 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

```

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

Istruzione 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;

```

Istruzione 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:

```

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

Conversione da Oracle ad Amazon RDS per Oracle

Alcuni aspetti da considerare durante la migrazione dello schema e del codice Oracle ad Amazon RDS per Oracle:

- AWS SCT può aggiungere oggetti di directory all'albero degli oggetti. Gli oggetti di directory sono strutture logiche che rappresentano ciascuna una directory fisica nel file system del server. È possibile utilizzare gli oggetti di directory con pacchetti come DBMS_LOB, UTL_FILE, DBMS_FILE_TRANSFER, la utility DATAPUMP e così via.
- AWS SCT supporta la conversione di tablespaces Oracle in un'istanza Amazon RDS per Oracle DB. Oracle memorizza i dati logicamente in spazi tabelle e fisicamente in file di dati associati allo spazio tabella corrispondente. In Oracle è possibile creare spazi tabella con i nomi dei file di dati. Amazon RDS supporta Oracle Managed Files (OMF) solo per file di dati, file di registro e file di controllo. AWS SCT crea i file di dati necessari durante la conversione.
- AWS SCT può convertire ruoli e privilegi a livello di server. Il motore di database Oracle utilizza la protezione basata sui ruoli. Un ruolo è una raccolta di privilegi che è possibile concedere o revocare a un utente. Un ruolo predefinito in Amazon RDS, denominato DBA, normalmente consente tutti i privilegi di amministrazione a un motore di database Oracle. I seguenti privilegi non sono disponibili per il ruolo DBA su un'istanza database di Amazon RDS utilizzando il motore Oracle:
 - Alter di database
 - Alter system

- Creazione di una directory
- Concessione di qualsiasi privilegio
- Concessione di qualsiasi ruolo
- Creazione di un processo esterno

È possibile concedere tutti gli altri privilegi a un ruolo utente Amazon RDS per Oracle, tra cui il filtro avanzato e i privilegi di colonna.

- AWS SCT supporta la conversione dei job Oracle in job eseguibili su Amazon RDS per Oracle. La conversione presenta alcuni limiti, inclusi i seguenti:
 - I processi eseguibili non sono supportati.
 - I processi di pianificazione che utilizzano il tipo di dati ANYDATA come un argomento non sono supportati.
- One Node RAC (Real Application Cluster) Oracle è un'opzione disponibile per Oracle Database Enterprise Edition che è stata introdotta con Oracle Database 11g Release 2. Amazon RDS per Oracle non supporta la caratteristica RAC. Per alta disponibilità, utilizzare Amazon RDS Implementazioni Multi-AZ.

In un'implementazione Multi-AZ, Amazon RDS effettua automaticamente il provisioning e mantiene una replica standby sincrona in un'altra zona di disponibilità. L'istanza database principale viene replicata in modo sincrono tra le zone di disponibilità in una replica di standby. Questa funzionalità fornisce ridondanza dei dati, elimina i blocchi I/O e riduce al minimo i picchi di latenza durante i backup di sistema.

- Oracle Spatial offre uno schema SQL e funzioni che semplificano le operazioni di storage, recupero, aggiornamento ed esecuzione di query per le raccolte di dati spaziali in un database Oracle. Oracle Locator offre funzionalità che sono solitamente richieste per il supporto di applicazioni Internet e wireless basate su servizi e di soluzioni GIS basate su partner. Oracle Locator è un sottoinsieme limitato di Oracle Spatial.

Per utilizzare le funzionalità di Oracle Spatial e Oracle Locator aggiungi l'opzione SPATIAL o l'opzione LOCATOR (che si escludono reciprocamente) al gruppo di opzioni dell'istanza database.

Esistono alcuni requisiti preliminari per l'utilizzo di Oracle Spatial e Oracle Locator su un'istanza database di Amazon RDS per Oracle:

- L'istanza deve utilizzare Oracle Enterprise Edition versione 12.1.0.2.v6 o successiva oppure 11.2.0.4.v10 o superiore.
- L'istanza deve essere all'interno di un cloud privato virtuale (VPC).

- L'istanza deve avere la classe dell'istanza database che può supportare la funzionalità Oracle. Ad esempio, Oracle Spatial non è supportato per le classi di istanza database db.m1.small, db.t1.micro, db.t2.micro o db.t2.small. Per ulteriori informazioni, vedere [Supporto delle classi di istanze DB per Oracle](#).
- L'istanza deve avere l'opzione Auto Minor Version Upgrade abilitata. Amazon RDS aggiorna l'istanza database al PSU Oracle più recente se esistono vulnerabilità di sicurezza con un punteggio CVSS score di 9+ o altre vulnerabilità di sicurezza annunciate. Per ulteriori informazioni, consulta la pagina

[Impostazioni per le istanze DB Oracle.](#)

- Se la versione dell'istanza database è 11.2.0.4.v10 o superiore, è necessario installare l'opzione XMLDB. Per ulteriori informazioni, consulta la pagina

[Oracle XML DB.](#)

- È necessaria una licenza per Oracle Spatial rilasciata da Oracle. Per ulteriori informazioni, consulta [Oracle Spatial and Graph](#) nella documentazione Oracle.
- Data Guard è incluso con Oracle Database Enterprise Edition. Per alta disponibilità, utilizzare la caratteristica Amazon RDS Implementazioni Multi-AZ.

In un'implementazione Multi-AZ, Amazon RDS effettua automaticamente il provisioning e mantiene una replica standby sincrona in un'altra zona di disponibilità. L'istanza database principale viene replicata in modo sincrono tra le zone di disponibilità in una replica di standby. Questa funzionalità fornisce ridondanza dei dati, elimina i blocchi I/O e riduce al minimo i picchi di latenza durante i backup di sistema.

- AWS SCT supporta la conversione di oggetti Oracle DBMS_SCHEDULER durante la migrazione ad Amazon RDS per Oracle. Il rapporto di AWS SCT valutazione indica se un oggetto della pianificazione può essere convertito. Per ulteriori informazioni sull'uso di oggetti di pianificazione con Amazon RDS, consulta la [documentazione di Amazon RDS](#).
- Per le conversioni da Oracle ad Amazon RDS per Oracle, è supportato DB Links. Un collegamento di database è un oggetto dello schema in un database che consente di accedere agli oggetti in un altro database. Non è necessario che gli altri database siano Oracle. Tuttavia, per accedere ai database diversi da Oracle, devi utilizzare gli Oracle Heterogeneous Services.

Una volta creato un collegamento di database, puoi utilizzarlo nelle istruzioni SQL per fare riferimento a tabelle, viste e oggetti PL/SQL nell'altro database. Per utilizzare un collegamento di database, aggiungi @dblink alla tabella, alla vista o al nome degli oggetti PL/SQL. Puoi eseguire

query su una tabella o vista nell'altro database con l'istruzione SELECT. Per ulteriori informazioni sull'utilizzo dei collegamenti di database Oracle, consulta la [documentazione di Oracle](#).

Per ulteriori informazioni sull'uso dei collegamenti di database con Amazon RDS, consulta la [documentazione di Amazon RDS](#).

- Il rapporto di AWS SCT valutazione fornisce le metriche del server per la conversione. Questi parametri sull'istanza Oracle includono quanto segue:
 - Capacità di calcolo e di memoria dell'istanza database di destinazione.
 - Funzionalità Oracle non supportate, come Real Application Clusters, non supportate da Amazon RDS.
 - Caricamento di lettura e scrittura su disco
 - Throughput del disco totale medio
 - Informazioni sul server quali il nome del server, il sistema operativo, il nome host e il set di caratteri.

Privilegi per RDS per Oracle come obiettivo

Per migrare ad Amazon RDS per Oracle, crea un utente privilegiato del database. È possibile utilizzare il seguente esempio di codice.

```
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;
```

Nell'esempio precedente, sostituisci *user_name* con il nome del tuo utente. Quindi, sostituisci *your_password* con una password sicura.

Limitazioni nella conversione da Oracle ad Amazon RDS per Oracle

Alcuni limiti da considerare durante la migrazione dello schema e del codice Oracle ad Amazon RDS per Oracle:

- Un ruolo predefinito in Amazon RDS, denominato DBA, normalmente consente tutti i privilegi di amministrazione a un motore di database Oracle. I seguenti privilegi non sono disponibili per il ruolo DBA su un'istanza database di Amazon RDS utilizzando il motore Oracle:
 - Alter di database
 - Alter system
 - Creazione di una directory
 - Concessione di qualsiasi privilegio
 - Concessione di qualsiasi ruolo
 - Creazione di un processo esterno

È possibile concedere tutti gli altri privilegi a un ruolo utente Oracle RDS.

- Amazon RDS per Oracle supporta l'auditing tradizionale, l'audit granulare utilizzando il pacchetto DBMS_FGA e Oracle Unified Auditing.
- Amazon RDS per Oracle non supporta la funzionalità CDC (Change Data Capture). Per eseguire il CDC durante e dopo una migrazione del database, usa AWS Database Migration Service.

Usare PostgreSQL come fonte per AWS SCT

È possibile utilizzare AWS SCT per convertire schemi, oggetti di codice del database e codice dell'applicazione da PostgreSQL ai seguenti obiettivi:

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

Per ulteriori informazioni, consulta le sezioni seguenti:

Argomenti

- [Privilegi per PostgreSQL come database di origine](#)
- [Connessione a PostgreSQL come sorgente](#)
- [Privilegi per MySQL come database di destinazione](#)

Privilegi per PostgreSQL come database di origine

I privilegi richiesti per PostgreSQL come sorgente sono i seguenti:

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

Connessione a PostgreSQL come sorgente

Utilizza la seguente procedura per connetterti al database di origine PostgreSQL con AWS Schema Conversion Tool.

Per connetterti a un database di origine PostgreSQL

1. NelAWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli PostgreSQL, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi sorgente.

3. Per Nome di connessione, immettere un nome per il database. AWS SCTvisualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del databaseAWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, utilizza le seguenti istruzioni:
 1. Per AWS Secret, scegli il nome del segreto.
 2. Scegli Popola per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere[Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine PostgreSQL, usa le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	<p>Immettere il nome Domain Name System (Domain Name System) o l'indirizzo IP del server di database di origine.</p> <p>Puoi connetterti al database PostgreSQL di origine utilizzando un protocollo di indirizzi IPv6. Per fare ciò, assicurati di utilizzare parentesi quadre per immettere l'indirizzo IP, come mostrato nell'esempio seguente.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 10px; text-align: center; margin-top: 10px;"><code>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</code></div>

Parametro	Operazione
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
Database	Immetti il nome del database PostgreSQL.
User name (Nome utente) e Password (Password)	<p>Immettere le credenziali del database per la connessione al server del database di origine.</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p>
Use SSL (Usa SSL)	<p>Scegli questa opzione per utilizzare Secure Sockets Layer (SSL) per la connessione al database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none">• Verifica certificato del server: seleziona questa opzione per verificare il certificato del server utilizzando un trust store.• Trust store: l'ubicazione di un trust store contenente certificati. Affinché questa posizione venga visualizzata nella sezione Impostazioni globali, assicurati di aggiungerla.
Memorizza la password	<p>AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Abilitando questa opzione potrai archiviare la password del database e connetterti in modo rapido al database senza dover inserire la password.</p>

Parametro	Operazione
Percorso del driver PostgreSQL	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

- Scegli Connessione di prova per verificare che sia inAWS SCT grado di connettersi al database di origine.
- Scegli Connect per connetterti al tuo database di origine.

Privilegi per MySQL come database di destinazione

I privilegi richiesti per MySQL come destinazione durante la migrazione da PostgreSQL sono i seguenti:

- CREA SU* . *
- MODIFICA DI * . *
- SALTA SU * . *
- INDICE SU * . *
- RIFERIMENTI SU * . *
- SELECT ON *.*
- CREA UNA VISTA SU* . *
- SHOW VIEW ON *.*
- ATTIVAZIONE SU * . *
- CREA UNA ROUTINE SU* . *
- MODIFICA LA ROUTINE SU * . *
- ESEGUI SU* . *
- INSERISCI, AGGIORNA SU AWS_POSTGRESQL_EXT. *

- INSERISCI, AGGIORNA, ELIMINA SU AWS_POSTGRESQL_EXT_DATA. *
- CREARE TABELLE TEMPORANEE SU AWS_POSTGRESQL_EXT_DATA. *

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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';
```

Nell'esempio precedente, sostituisci *user_name* con il nome del tuo utente. Quindi, sostituisci *your_password* con una password sicura.

Per utilizzare Amazon RDS for MySQL o Aurora MySQL come destinazione, imposta il `lower_case_table_names` parametro su 1. Questo valore indica che il server MySQL gestisce gli identificatori di nomi di oggetti come tabelle, indici, trigger e database senza distinzione tra maiuscole e minuscole. Se hai attivato la registrazione binaria nell'istanza di destinazione, imposta il `log_bin_trust_function_creators` parametro su 1. In questo caso, non è necessario utilizzare le `DETERMINISTIC` SQL caratteristiche `READS SQL DATA` o per creare funzioni memorizzate. Per configurare questi parametri, crea un gruppo di parametri di database o modificane uno esistente.

Utilizzo di SAP ASE (Sybase ASE) come fonte per AWS SCT

È possibile utilizzarli AWS SCT per convertire schemi, oggetti di codice del database e codice dell'applicazione da SAP (Sybase) Adaptive Server Enterprise (ASE) alle seguenti destinazioni:

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

Per ulteriori informazioni, consulta le sezioni seguenti:

Argomenti

- [Privilegi per SAP ASE come database di origine](#)
- [Connessione a SAP ASE \(Sybase\) come sorgente](#)
- [Privilegi per MySQL come database di destinazione](#)
- [Impostazioni di conversione da SAP ASE a MySQL](#)
- [Privilegi per PostgreSQL come database di destinazione](#)
- [Impostazioni di conversione da SAP ASE a PostgreSQL](#)

Privilegi per SAP ASE come database di origine

Per utilizzare un database SAP ASE come origine, è necessario creare un utente del database e concedere le autorizzazioni. Per fare ciò, completa la procedura seguente.

Creazione e configurazione di un utente del database

1. Connettiti al database di origine.
2. Creazione di un utente del database con i seguenti comandi. Fornire una password per il nuovo utente.

```
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. Per ogni database che intendi migrare, concedi i seguenti privilegi.

```
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
```

Connessione a SAP ASE (Sybase) come sorgente

Utilizza la seguente procedura per connetterti al database di origine SAP ASE con AWS Schema Conversion Tool.

Per connettersi a un database di origine SAP ASE


1. Nel AWS Schema Conversion Tool, scegli **Aggiungi fonte**.
2. Scegli **SAP ASE**, quindi scegli **Avanti**.

Viene visualizzata la finestra di dialogo **Aggiungi fonte**.

3. In **Name (Nome connessione)**, immettere un nome per il database. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, utilizza le seguenti istruzioni:
 1. Per **AWS Secret**, scegli il nome del segreto.
 2. Scegli **Popula** per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine SAP ASE, utilizzare le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Immettere il nome Domain Name System (DNS) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
Database	Immettere il nome del database SAP ASE.
User name (Nome utente) e Password (Password)	<p>Immettere le credenziali del database per connetterti al server del database di origine.</p> <div data-bbox="656 716 1507 1220" style="border: 1px solid #add8e6; border-radius: 15px; padding: 10px; margin-top: 10px;"> <p> Note</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p> </div>
Use SSL (Usa SSL)	<p>Scegliete questa opzione per utilizzare il protocollo SSL (Secure Sockets Layer) per connetterti al database. Fornisci le seguenti informazioni aggiuntive, se applicabile, nella scheda SSL:</p> <ul style="list-style-type: none"> • Verifica certificato del server: seleziona questa opzione per verificare il certificato del server utilizzando un trust store. • Trust store: la posizione di un trust store contenente certificati.

Parametro	Operazione
Memorizza la password	AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Abilitando questa opzione potrai archiviare la password del database e connetterti in modo rapido al database senza dover inserire la password.
Percorso del driver SAP ASE	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

- Scegli Connessione di prova per verificare che sia inAWS SCT grado di connettersi al database di origine.
- Scegli Connect per connetterti al tuo database di origine.

Privilegi per MySQL come database di destinazione

I privilegi richiesti per MySQL come destinazione sono i seguenti:

- CREA SU* . *
- MODIFICA DI (*) . *
- SALTA SU * . *
- INDICE SU * . *
- RIFERIMENTI SU * . *
- SELECT ON *.*
- CREA UNA VISTA SU* . *
- SHOW VIEW ON *.*
- ATTIVAZIONE DI * . *
- CREA UNA ROUTINE SU* . *

- MODIFICA LA ROUTINE SU *.*
- ESEGUI SU *.*
- INSERISCI, AGGIORNA SU AWS_SAPASE_EXT.*
- CREARE TABELLE TEMPORANEE SU AWS_SAPASE_EXT.*

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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';
```

Nell'esempio precedente, sostituisci *user_name* con il nome del tuo utente. Quindi, sostituisci *your_password* con una password sicura.

Per utilizzare Amazon RDS for MySQL o Aurora MySQL come destinazione, imposta il `lower_case_table_names` parametro su 1. Questo valore indica che il server MySQL gestisce gli identificatori di nomi di oggetti come tabelle, indici, trigger e database senza distinzione tra maiuscole e minuscole. Se hai attivato la registrazione binaria nell'istanza di destinazione, imposta il `log_bin_trust_function_creators` parametro su 1. In questo caso, non è necessario utilizzare le `DETERMINISTIC` SQL caratteristiche `READS SQL DATA` o per creare funzioni memorizzate. Per impostare questi parametri, crea un gruppo di parametri di database o modificane uno esistente.

Impostazioni di conversione da SAP ASE a MySQL

Per modificare le impostazioni di conversione da SAP ASE a MySQL, scegli Impostazioni, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli SAP ASE, quindi scegli SAP ASE — MySQL o SAP ASE — Amazon Aurora (compatibile con MySQL). AWS SCT visualizza tutte le impostazioni disponibili per la conversione da SAP ASE a PostgreSQL.

Le impostazioni di conversione da SAP ASE a MySQL AWS SCT includono le opzioni per quanto segue:

- Per limitare il numero di commenti con azioni nel codice convertito.

Per Aggiungi commenti nel codice convertito per le azioni di gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per utilizzare i nomi esatti degli oggetti del database di origine nel codice convertito.

Per impostazione predefinita, AWS SCT converte i nomi degli oggetti, delle variabili e dei parametri del database in lettere minuscole. Per mantenere le maiuscole e minuscole per questi nomi, selezionare Considera i nomi degli oggetti del database di origine con distinzione tra maiuscole e minuscole. Scegliete questa opzione se utilizzate nomi di oggetti con distinzione tra maiuscole e minuscole nel server del database SAP ASE di origine.

Privilegi per PostgreSQL come database di destinazione

Per utilizzare PostgreSQL come destinazione, è necessario il CREATE ON DATABASE privilegio. Assicurati di concedere questo privilegio per ogni database PostgreSQL di destinazione.

Per utilizzare i sinonimi pubblici convertiti, modifica il percorso di ricerca predefinito del database in "\$user", public_synonyms, public.

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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;
```

Nell'esempio precedente, sostituisci *user_name* con il nome del tuo utente. Quindi, sostituisci *db_name* con il nome del tuo database di destinazione. Infine, sostituisci *your_password* con una password sicura.

In PostgreSQL, solo il proprietario dello schema o `unsuperuser` può eliminare uno schema. Il proprietario può eliminare uno schema e tutti gli oggetti che questo schema include anche se il proprietario dello schema non possiede alcuni dei suoi oggetti.

Quando si utilizzano utenti diversi per convertire e applicare schemi diversi al database di destinazione, è possibile ricevere un messaggio di errore quando non è possibile eliminare uno schema. Per evitare questo messaggio di errore, usa il `superuser` ruolo.

Impostazioni di conversione da SAP ASE a PostgreSQL

Per modificare le impostazioni di conversione da SAP ASE a PostgreSQL, scegli Impostazioni, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli SAP ASE, quindi scegli SAP ASE — PostgreSQL o SAP ASE — Amazon Aurora (compatibile con PostgreSQL). AWS SCT visualizza tutte le impostazioni disponibili per la conversione da SAP ASE a PostgreSQL.

Le impostazioni di conversione da SAP ASE a PostgreSQL AWS SCT includono le opzioni per quanto segue:

- Per limitare il numero di commenti con azioni nel codice convertito.

Per Aggiungere commenti nel codice convertito per le azioni di gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per definire il modello da utilizzare per i nomi degli schemi nel codice convertito. Per il modello di generazione del nome dello schema, scegli una delle seguenti opzioni:
 - `<source_db>`— Utilizza il nome del database SAP ASE come nome dello schema in PostgreSQL.
 - `<source_schema>`— Utilizza il nome dello schema SAP ASE come nome dello schema in PostgreSQL.

- `<source_db><schema>`— Utilizza una combinazione del database SAP ASE e dei nomi degli schemi come nome dello schema in PostgreSQL.
- Per utilizzare i nomi esatti degli oggetti del database di origine nel codice convertito.

Per impostazione predefinita, AWS SCT converte i nomi degli oggetti, delle variabili e dei parametri del database in lettere minuscole. Per mantenere le maiuscole e minuscole per questi nomi, selezionare Considera i nomi degli oggetti del database di origine con distinzione tra maiuscole e minuscole. Scegliete questa opzione se utilizzate nomi di oggetti con distinzione tra maiuscole e minuscole nel server del database SAP ASE di origine.

Per le operazioni con distinzione tra maiuscole e minuscole, AWS SCT può evitare la conversione dei nomi degli oggetti del database in lettere minuscole. A tale scopo, seleziona Evita la conversione in lettere minuscole per le operazioni con distinzione tra maiuscole e minuscole.

- Per consentire l'uso di indici con lo stesso nome in tabelle diverse in SAP ASE.

In PostgreSQL, tutti i nomi degli indici utilizzati nello schema devono essere univoci. Per assicurarti che AWS SCT generi nomi univoci per tutti i tuoi indici, seleziona Genera nomi univoci per gli indici.

Utilizzo di Microsoft SQL Server come fonte per AWS SCT

È possibile utilizzare AWS SCT per convertire schemi, oggetti di codice del database e codice dell'applicazione da SQL Server alle seguenti destinazioni:

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

Note

AWS SCT non supporta l'utilizzo di Amazon RDS per SQL Server come sorgente.

È possibile utilizzare AWS SCT per creare un rapporto di valutazione per la migrazione di schemi, oggetti di codice del database e codice applicativo da SQL Server a Babelfish per Aurora PostgreSQL, come descritto di seguito.

Argomenti

- [Privilegi per Microsoft SQL Server come fonte](#)
- [Utilizzo dell'autenticazione di Windows quando si utilizza Microsoft SQL Server come origine](#)
- [Connessione a SQL Server come origine](#)
- [Conversione da SQL Server a MySQL](#)
- [Conversione da SQL Server PostgreSQL](#)
- [Conversione di SQL Server in Amazon RDS per SQL Server](#)

Privilegi per Microsoft SQL Server come fonte

I privilegi richiesti per Microsoft SQL Server come origine sono i seguenti:

- VIEW DEFINITION
- VIEW DATABASE STATE

Il `VIEW DEFINITION` privilegio consente agli utenti con accesso pubblico di visualizzare le definizioni degli oggetti. AWS SCT utilizza il `VIEW DATABASE STATE` privilegio per verificare le funzionalità dell'edizione SQL Server Enterprise.

Ripeti l'assegnazione per ogni database di cui desideri convertire lo schema.

Inoltre, concedi i seguenti privilegi su master banca dati:

- VIEW SERVER STATE
- VISUALIZZA QUALSIASI DEFINIZIONE

AWS SCT utilizza il `VIEW SERVER STATE` privilegio per raccogliere le impostazioni e la configurazione del server. Assicurati di concedere il `VIEW ANY DEFINITION` privilegio di visualizzare gli endpoint.

Per leggere informazioni su Microsoft Analysis Services, eseguire il comando seguente da master database.

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

Nell'esempio precedente, sostituire `<user_name>` segnaposto con il nome dell'utente a cui hai concesso i privilegi in precedenza.

Per leggere informazioni su SQL Server Agent, aggiungi il tuo utente al `SQLAgentUser` ruolo. Eseguire il comando seguente nel msdb banca dati.

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

Nell'esempio precedente, sostituire `<SQLAgentRole>` segnaposto con il nome del ruolo di SQL Server Agent. Quindi sostituire il `<user_name>` segnaposto con il nome dell'utente a cui hai concesso i privilegi in precedenza. Per ulteriori informazioni, consulta la pagina [Aggiungere un utente a SQLAgentUser ruolo](#) nel Guida per l'utente di Amazon RDS.

Per rilevare la spedizione in log, concedi il `SELECT` on `dbo.log_shipping_primary_databases` privilegio sul msdb banca dati.

Per utilizzare l'approccio di notifica della replica DDL, concedi `RECEIVE ON <schema_name>.<queue_name>` privilegio sui database di origine. In questo esempio, sostituire i `<schema_name>` segnaposto con il nome dello schema del database. Quindi, sostituisci il `<queue_name>` segnaposto con il nome di una tabella di coda.

Utilizzo dell'autenticazione di Windows quando si utilizza Microsoft SQL Server come origine

Se la tua applicazione viene eseguita su un'intranet basata su Windows, puoi utilizzare l'autenticazione di Windows per l'accesso al database. Per accedere al database SQL Server, l'autenticazione di Windows utilizza l'identità Windows corrente stabilita sul thread del sistema operativo. Puoi quindi mappare l'identità Windows a un database SQL Server e alle autorizzazioni. Per connettersi a SQL Server utilizzando l'autenticazione di Windows, devi specificare l'identità Windows utilizzata dall'applicazione sta utilizzando. Inoltre, devi concedere l'accesso all'identità Windows al database SQL Server.

SQL Server prevede due modalità di accesso: modalità autenticazione di Windows e modalità mista. La modalità autenticazione di Windows abilita l'autenticazione di Windows e disabilita l'autenticazione di SQL Server. La modalità mista abilita sia l'autenticazione di Windows che l'autenticazione di SQL

Server. L'autenticazione di Windows è sempre disponibile e non può essere disattivata. Per ulteriori informazioni sull'autenticazione di Windows, consulta la documentazione di Microsoft Windows.

Di seguito è riportato un possibile esempio di creazione di un utente in TEST_DB

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

Utilizzo dell'autenticazione di Windows con una connessione JDBC

Il driver JDBC non supporta l'autenticazione di Windows se viene utilizzato su sistemi operativi diversi da Windows. Le credenziali di autenticazione di Windows, ad esempio nome utente e password, non vengono specificate automaticamente durante la connessione a SQL Server da sistemi operativi non Windows. In questi casi, le applicazioni devono invece utilizzare l'autenticazione di SQL Server.

Nella stringa di connessione a JDBC, per connettersi utilizzando l'autenticazione di Windows deve essere specificato il parametro `integratedSecurity`. Il driver JDBC supporta l'autenticazione integrata di Windows su sistemi operativi Windows tramite il parametro della stringa di connessione `integratedSecurity`.

Per utilizzare l'autenticazione integrata

1. Installa il driver JDBC.
2. Copia il file `sqljdbc_auth.dll` in una directory nel percorso di sistema di Windows sul computer in cui è installato il driver JDBC.

I file `sqljdbc_auth.dll` vengono installati nella seguente posizione:

```
<directory di installazione>\sqljdbc_<versione>\<lingua>\auth\
```

Quando tenti di stabilire una connessione al database SQL Server utilizzando l'autenticazione di Windows, potresti ricevere l'errore "Il driver non è configurato per l'autenticazione integrata". Questo problema può essere risolto tramite le operazioni seguenti:

- dichiarare due variabili che puntano al percorso di installazione del JDBC:

```
variable name: SQLJDBC_HOME; variable value: D:\lib\JDBC4.1\enu(dove esiste il file sqljdbc4.jar);
```

variable name: SQLJDBC_AUTH_HOME; variable value: D:\lib\JDBC4.1\enu\auth\x86(se si utilizza un sistema operativo a 32 bit) oppureD:\lib\JDBC4.1\enu\auth\x64(se utilizzi un sistema operativo a 64 bit). È qui che si trova sqljdbc_auth.dll si trova.

- Copia sqljdbc_auth.dll nella cartella in cui è in esecuzione il tuo JDK/JRE. è possibile eseguire la copia nella cartella lib, bin, ecc. Ad esempio, è possibile copiare nella cartella seguente.

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

- Assicurati che nella cartella della libreria JDBC sia presente solo il file SQLJDBC4.jar. Rimuovi tutti gli altri file sqljdbc*.jar da quella cartella (o copiali in un'altra cartella). Se state aggiungendo il driver come parte del programma, assicuratevi di aggiungere solo SQLJDBC4.jar come driver da utilizzare.
- copiare il file sqljdbc_auth.dll nella cartella dell'applicazione.

Note

Se è in esecuzione una Java Virtual Machine (JVM) a 32 bit, utilizza il file sqljdbc_auth.dll nella cartella x86 anche se la versione del sistema operativo è x64. Se è in esecuzione una JVM a 64 bit su un processore x64, utilizza il file sqljdbc_auth.dll nella cartella x64.

Quando effettui la connessione a un database SQL Server, puoi scegliere Autenticazione Windows o Autenticazione SQL Server per l'opzione Autenticazione.

Connessione a SQL Server come origine

Utilizza la seguente procedura per connetterti al database di origine Microsoft SQL Server con AWS Schema Conversion Tool.

Per connettersi a un database di origine Microsoft SQL Server

1. NelAWS Schema Conversion Tool, scegliAggiungi fonte.
2. ScegliMicrosoft SQL Server, quindi scegliAvanti.

LaAggiungi fonteViene visualizzata la finestra di dialogo.

3. PerNome della connessione, inserisci un nome per il database.AWS SCTvisualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database diAWS Secrets Manageroppure inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, utilizzare le seguenti istruzioni:
 1. PerSegreto AWS, scegli il nome del segreto.
 2. ScegliPopolaper inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database da Secrets Manager.

Per informazioni sull'utilizzo delle credenziali database da Secrets Manager, vedere[Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni sulla connessione al database di origine di Microsoft SQL Server, utilizzare le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	<p>Immetti il nome Domain Name Service (DNS) o l'indirizzo IP del server del database di origine.</p> <p>Puoi connetterti al database di SQL Server da dall'istanza database di SQL Server utilizzando un protocollo di indirizzi IPv6. A tale scopo, assicurati di utilizzare parentesi quadre per inserire l'indirizzo IP, come illustrato nell'esempio seguente.</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>
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
Instance name (Nome dell'istanza)	Immetti il nome dell'istanza per il database SQL Server. Per trovare il nome dell'istanza, lancia la query <code>SELECT @@servername;</code> sul database SQL Server.
Autenticazione	Scegli il tipo di autenticazione traAutenticazione WindowseAutenticazione di SQL Server.

Parametro	Operazione
User name (Nome utente) e Password (Password)	<p>Inserisci le credenziali database per connetterti al database server di origine.</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password del database di origine, AWS SCT per impostazione predefinita, non memorizza la password. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p>
Use SSL (Usa SSL)	<p>Scegli questa opzione per utilizzare Secure Sockets Layer (SSL) per connettersi al database. Fornisci le seguenti informazioni aggiuntive, se applicabili, su SSL scheda:</p> <ul style="list-style-type: none">• Certificato del server di fiducia: seleziona questa opzione per considerare attendibile il certificato del server.• Trust Store: la posizione di un archivio attendibile contenente certificati. Affinché questa posizione appaia nell'impostazioni globali, assicurati di aggiungerla.
Memorizza la password	<p>AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Abilitando questa opzione potrai archiviare la password del database e connetterti in modo rapido al database senza dover inserire la password.</p>

Parametro	Operazione
Sql Server Driver Path (Percorso driver Sql Server)	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>
Libreria di autenticazione di Windows	<p>Immettete il percorso del file <code>sqljdbc_auth.dll</code>. Per impostazione predefinita, questo file viene installato nella seguente posizione:</p> <p><i><installation directory of the JDBC driver>sqljdbc_<version> \<language> \auth\</i></p>

- Scegli **Connessione di prova** per verificare che AWS SCT può connettersi al database di origine.
- Scegli **Connect** per connettersi al database di origine.

Conversione da SQL Server a MySQL

Per emulare le funzioni del database di Microsoft SQL Server nel codice MySQL convertito, usa il pacchetto di estensione SQL Server to MySQL in AWS SCT. Per ulteriori informazioni relative ai pacchetti di estensione, vedere [Utilizzo dei pacchetti di AWS SCT estensione](#).

Argomenti

- [Privilegi per MySQL come database di destinazione](#)
- [Impostazioni di conversione da SQL Server a MySQL](#)
- [Considerazioni sulla migrazione](#)

Privilegi per MySQL come database di destinazione

I privilegi richiesti per MySQL come destinazione sono i seguenti:

- CREA SU* . *
- MODIFICA SU* . *
- SCENDI SU* . *
- INDICE SU* . *
- RIFERIMENTI SU* . *
- SELECT ON *.*
- CREA UNA VISTA SU* . *
- SHOW VIEW ON *.*
- SELEZIONA SU* . *
- CREA ROUTINE SU* . *
- MODIFICA LA ROUTINE SU* . *
- ESEGUI SU* . *
- INSERISCI, AGGIORNA SU AWS_SQLSERVER_EXT. *
- INSERISCI, AGGIORNA, ELIMINA SU AWS_SQLSERVER_EXT_DATA. *
- CREA TABELLE TEMPORANEE SU AWS_SQLSERVER_EXT_DATA. *

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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';
```


Nell'esempio precedente, sostituire *nome_utente* con il nome del tuo utente. Quindi, sostituisci *tua_password* con una password sicura.

Se utilizzi un database MySQL versione 5.7 o precedente come destinazione, esegui il comando seguente. Per i database MySQL versione 8.0 e successive, questo comando è obsoleto.

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

Per utilizzare Amazon RDS for MySQL o Aurora MySQL come destinazione, imposta `lower_case_table_names` su 1. Questo valore indica che il server MySQL gestisce gli identificatori di nomi di oggetti come tabelle, indici, trigger e database senza distinzione tra maiuscole e minuscole. Se hai attivato la registrazione binaria nell'istanza di destinazione, imposta `log_bin_trust_function_creators` su 1. In questo caso, non è necessario utilizzare `DETERMINISTIC, READS SQL DATA` o `NO SQL` caratteristiche per creare funzioni memorizzate. Per configurare questi parametri, crea un nuovo gruppo di parametri database o modifica un gruppo di parametri database.

Impostazioni di conversione da SQL Server a MySQL

Per modificare le impostazioni di conversione da SQL Server a MySQL, in AWS SCT scegliere impostazioni e quindi scegliere Impostazioni di conversione. Dall'elenco superiore, scegli SQL Server e quindi scegli SQL Server — MySQL. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da SQL Server a MySQL.

Impostazioni di conversione da SQL Server a MySQL in AWS SCT include le seguenti opzioni relative ai campi seguenti:

- Per limitare il numero di commenti con azioni nel codice convertito.

Per aggiungere commenti nel codice convertito per le azioni di gravità selezionata o superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata o superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegliete Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per consentire al database SQL Server di origine di archiviare l'output di EXEC in una tabella. AWS SCT crea tabelle temporanee e una procedura aggiuntiva per emulare questa funzionalità. Per utilizzare questa emulazione, seleziona Crea routine aggiuntive per la gestione di set di dati aperti.

Considerazioni sulla migrazione

Considerate questi aspetti durante la migrazione di uno schema di SQL Server su MySQL:

- MySQL non supporta `IMERGE` dichiarazione. Tuttavia, AWS SCT può emulare `IMERGE` dichiarazione durante la conversione utilizzando `INSERT ON DUPLICATE KEY` clausola e `UPDATE FROM` and `DELETE FROM` dichiarazioni.

Per una corretta emulazione usando `INSERT ON DUPLICATE KEY`, assicurati che esista un vincolo univoco o una chiave primaria sul database MySQL di destinazione.

- È possibile utilizzare `GOTO` istruzione e un'etichetta per modificare l'ordine in cui vengono eseguite le istruzioni. Qualsiasi istruzione Transact-SQL che segue un `GOTO` le istruzioni vengono ignorate e l'elaborazione continua dall'etichetta. È possibile utilizzare `GOTO` istruzioni ed etichette ovunque all'interno di una procedura, batch o blocco di istruzioni. Puoi anche nidificare `GOTO` dichiarazioni.

MySQL non usa `GOTO` dichiarazioni. Quando AWS SCT converte il codice che contiene un `GOTO` dichiarazione, converte l'istruzione in uso a `BEGIN...END` o `LOOP...END LOOP` dichiarazione. Puoi trovare esempi di come AWS SCT converte `GOTO` dichiarazioni nella tabella seguente.

Istruzione SQL Server	Istruzione 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;</pre>	<pre>BEGIN statement1;</pre>

Istruzione SQL Server	Istruzione MySQL
<pre> label1: statement2; GOTO label1; statement3; statement4; END </pre>	<pre> label1: LOOP statement2; ITERATE label1; LEAVE label1; END LOOP; statement3; statement4; END </pre>
<pre> BEGIN statement1; label1: statement2; statement3; statement4; END </pre>	<pre> BEGIN statement1; label1: BEGIN statement2; statement3; statement4; END; END </pre>

- MySQL non supporta funzioni con valori di tabella a più istruzioni. AWS SCT simula funzioni con valori di tabella durante una conversione creando tabelle temporanee e riscrivendo istruzioni per utilizzare queste tabelle temporanee.

Conversione da SQL Server PostgreSQL

È possibile utilizzare il pacchetto di estensione da SQL Server a PostgreSQL in AWS SCT. Questo pacchetto di estensione emula le funzioni del database di SQL Server nel codice PostgreSQL convertito. Usa il pacchetto di estensione SQL Server to PostgreSQL per emulare SQL Server

Agent e SQL Server Database Mail. Per ulteriori informazioni relative ai pacchetti di estensione, vedere [Utilizzo dei pacchetti di AWS SCT estensione](#).

Argomenti

- [Privilegi per PostgreSQL come database di destinazione](#)
- [Impostazioni di conversione da SQL Server a PostgreSQL](#)
- [Conversione di partizioni SQL Server in partizioni PostgreSQL versione 10](#)
- [Considerazioni sulla migrazione](#)
- [Utilizzare unAWS SCTpacchetto di estensioni per emulare SQL Server Agent in PostgreSQL](#)
- [Utilizzare unAWS SCTpacchetto di estensione per emulare SQL Server Database Mail in PostgreSQL](#)

Privilegi per PostgreSQL come database di destinazione

Per usare PostgreSQL come destinazione, AWS SCT richiede `CREATE ON DATABASE` privilegio. Assicurati di concedere questo privilegio per ogni database PostgreSQL di destinazione.

Per utilizzare i sinonimi pubblici convertiti, modifica il percorso di ricerca predefinito del database in `"$user", public_synonyms, public`.

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere i privilegi.

```
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;
```

Nell'esempio precedente, sostituire *nome_utente* con il nome del tuo utente. Quindi, sostituisci *db_name* con il nome del database di destinazione. Infine, sostituisci *your_password* con una password sicura.

In PostgreSQL, solo il proprietario dello schema o un `superuser` può eliminare uno schema. Il proprietario può eliminare uno schema e tutti gli oggetti inclusi in questo schema anche se il proprietario dello schema non possiede alcuni dei suoi oggetti.

Quando utilizzi utenti diversi per convertire e applicare schemi diversi al database di destinazione, puoi ricevere un messaggio di errore quando AWS SCT non posso eliminare lo schema. Per evitare questo messaggio di errore, utilizza `superuser` ruolo.

Impostazioni di conversione da SQL Server a PostgreSQL

Per modificare le impostazioni di conversione da SQL Server a PostgreSQL, scegli **Impostazioni**, quindi seleziona **Impostazioni di conversione**. Dall'elenco superiore, scegli **SQL Server**, quindi seleziona **SQL Server — PostgreSQL**. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da SQL Server a PostgreSQL.

Impostazioni di conversione da SQL Server a PostgreSQL in AWS SCT include le seguenti opzioni relative ai campi seguenti:

- Per limitare il numero di commenti con azioni nel codice convertito.

Per aggiungere commenti nel codice convertito per le azioni di gravità selezionata o superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata o superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli **Solo errori**. Per includere commenti per tutte le azioni nel codice convertito, scegli **Tutti i messaggi**.

- Per consentire l'utilizzo di indici con lo stesso nome in tabelle diverse in SQL Server.

In PostgreSQL, tutti i nomi di indice utilizzati nello schema devono essere univoci. Per essere sicuri che AWS SCT genera nomi univoci per tutti i tuoi indici, seleziona **Genera nomi univoci per gli indici**.

- Per convertire le procedure di SQL Server in funzioni PostgreSQL.

PostgreSQL versione 10 e precedenti non supportano le procedure. Per i clienti che non hanno familiarità con l'uso delle procedure in PostgreSQL, AWS SCT può convertire le procedure in funzioni. Per fare ciò, selezionare **Convertire le procedure in funzioni**.

- Per emulare l'output di EXEC in una tabella.

Il database SQL Server di origine può memorizzare l'output di EXEC in una tabella. AWS SCT crea tabelle temporanee e una procedura aggiuntiva per emulare questa funzionalità. Per utilizzare questa emulazione, selezionare **Crea routine aggiuntive per la gestione di set di dati aperti**.

- Per definire il modello da utilizzare per i nomi degli schemi nel codice convertito. Per **Modello di generazione del nome dello schema**, scegli una delle seguenti opzioni:
 - **<source_db>**— Utilizza il nome del database SQL Server come nome dello schema in PostgreSQL.
 - **<source_schema>**— Utilizza il nome dello schema di SQL Server come nome dello schema in PostgreSQL.

- `<source_db>_<schema>`— Utilizza una combinazione del database di SQL Server e dei nomi dello schema come nome dello schema in PostgreSQL.
- Per mantenere le lettere maiuscole e minuscole dei nomi degli oggetti di origine.

Per evitare la conversione dei nomi degli oggetti in lettere minuscole, selezionate **Evita di utilizzare lettere minuscole** per le operazioni che fanno distinzione tra maiuscole. Questa opzione si applica solo quando si attiva l'opzione di distinzione tra maiuscole e minuscole nel database di destinazione.

- Per mantenere i nomi dei parametri lontani dal database di origine.

Per aggiungere virgolette doppie ai nomi dei parametri nel codice convertito, selezionate **Mantieni i nomi dei parametri originali**.

Conversione di partizioni SQL Server in partizioni PostgreSQL versione 10

Quando converti un database Microsoft SQL Server in Amazon Aurora PostgreSQL Compatible Edition (Aurora PostgreSQL) o Amazon Relational Database Service for PostgreSQL (Amazon RDS per PostgreSQL), tieni presente quanto segue.

In SQL Server, è necessario creare partizioni con le funzioni di partizione. Quando esegui la conversione da una tabella porzionata SQL Server a una tabella partizionata PostgreSQL versione 10, devi essere consapevole di alcuni problemi potenziali:

- SQL Server consente di partizionare una tabella utilizzando una colonna senza un vincolo NOT NULL. In questo caso, tutti i valori NULL vanno alla partizione più a sinistra. PostgreSQL non supporta valori NULL per il partizionamento RANGE.
- SQL Server consente di creare chiavi primarie e univoche per le tabelle partizionate. Per PostgreSQL, è necessario creare chiavi primarie o univoche per ogni partizione direttamente. Pertanto, il vincolo PRIMARY o UNIQUE KEY deve essere rimosso dalla tabella padre durante la migrazione a PostgreSQL. I nomi chiave risultanti assumono il formato `<original_key_name>_<partition_number>`.
- SQL Server consente di creare un vincolo di chiave esterna da e verso tabelle partizionate. PostgreSQL non supporta le chiavi esterne che fanno riferimento a tabelle partizionate. Inoltre, PostgreSQL non supporta i riferimenti delle chiavi esterne da una tabella partizionata a un'altra tabella.
- SQL Server consente di creare indici per le tabelle partizionate. Per PostgreSQL, è necessario creare un indice per ogni partizione direttamente. In questo modo, gli indici devono essere rimossi

dalle loro tabelle padre durante la migrazione a PostgreSQL. I nomi degli indici risultanti avranno il formato `<original_index_name>_<partition_number>`.

- PostgreSQL non supporta indici partizionati.

Considerazioni sulla migrazione

Alcuni aspetti da considerare durante la migrazione di uno schema di SQL Server a PostgreSQL:

- In PostgreSQL, tutti i nomi di oggetti in uno schema devono essere univoci, inclusi gli indici. I nomi degli indici devono essere univoci nello schema della tabella di base. In SQL Server, un nome indice può essere lo stesso per tabelle diverse.

Per garantire l'unicità dei nomi degli indici, AWS SCT offre la possibilità di generare nomi di indice univoci se i nomi degli indici non sono univoci. A questo scopo scegli l'opzione **Genera nomi indici univoci** nelle proprietà del progetto. Per impostazione predefinita, questa opzione è abilitata. Se questa opzione è attiva, i nomi indice univoci vengono creati utilizzando il formato `IX_table_name_index_name`. Se questa opzione è disattivata, i nomi indice non vengono modificati.

- Un'istruzione `GOTO` e un'etichetta possono essere utilizzati per modificare l'ordine in cui vengono eseguite le istruzioni. Qualsiasi istruzione Transact-SQL che segue un'istruzione `GOTO` viene ignorata e l'elaborazione continua a livello di etichetta. Le istruzioni e le etichette `GOTO` possono essere utilizzate ovunque all'interno di una procedura, di un batch o di un blocco dell'istruzione. Le istruzioni `GOTO` possono anche essere nidificate.

PostgreSQL non utilizza le istruzioni `GOTO`. Quando AWS SCT converte il codice che contiene un'istruzione `GOTO`, converte l'istruzione per utilizzare un'istruzione `BEGIN... END` o `LOOP... END LOOP`. Puoi trovare esempi di come AWS SCT converte le istruzioni `GOTO` nella tabella seguente.

Istruzioni `GOTO` SQL Server e istruzioni PostgreSQL convertite

Istruzione SQL Server	Istruzione PostgreSQL
<pre>BEGIN statement1; GOTO label1; statement2; label1:</pre>	<pre>BEGIN label1: BEGIN statement1; EXIT label1; statement2;</pre>

Istruzione SQL Server	Istruzione PostgreSQL
<pre>Statement3; END</pre>	<pre>.... END; Statement3; END</pre>
<pre>BEGIN statement1; label1: statement2; GOTO label1; statement3; statement4; END</pre>	<pre>BEGIN statement1; label1: LOOP statement2; CONTINUE label1; EXIT label1; END LOOP; statement3; statement4; END</pre>
<pre>BEGIN statement1; label1: statement2; statement3; statement4; END</pre>	<pre>BEGIN statement1; label1: BEGIN statement2; statement3; statement4; END; END</pre>

- PostgreSQL non supporta un'istruzione MERGE. AWS SCT emula il comportamento di un'istruzione MERGE nei seguenti modi:
 - Tramite la costruzione INSERT ON CONFLICT.
 - Utilizzando l'istruzione UPDATE FROM DML, ad esempio MERGE senza una clausola WHEN NOT MATCHED.
 - Utilizzando CURSOR, ad esempio con una clausola MERGE with DELETE o utilizzando un'istruzione di condizione MERGE ON complessa.
- AWS SCT può aggiungere trigger di database all'albero degli oggetti quando Amazon RDS è la destinazione.
- AWS SCT può aggiungere trigger a livello di server all'albero degli oggetti quando Amazon RDS è la destinazione.
- SQL Server crea e gestisce automaticamente `deleted` e `inserted` tabelle. È possibile utilizzare queste tabelle temporanee residenti in memoria per testare gli effetti di determinate modifiche ai dati e impostare le condizioni per le azioni di attivazione DML. AWS SCT può convertire l'utilizzo di queste tabelle all'interno delle istruzioni trigger DML.
- AWS SCT può aggiungere server collegati all'albero degli oggetti quando Amazon RDS è la destinazione.
- Durante la migrazione da Microsoft SQL Server a PostgreSQL, la funzione `SUSER_SNAME` integrata viene convertita come segue:
 - `SUSER_SNAME` - Restituisce il nome di accesso associato a un numero di identificazione di sicurezza (SID).
 - `SUSER_SNAME(<server_user_sid>)` – Non supportata.
 - `SUSER_SNAME() CURRENT_USER` – Restituisce il nome utente del contesto di esecuzione corrente.
 - `SUSER_SNAME(NULL)` – Restituisce NULL.
- La conversione di funzioni valutate a livello di tabella è supportata. Le funzioni valutate a livello di tabella restituiscono una tabella e possono prendere il posto di una tabella in una query.
- `PATINDEX` restituisce la posizione iniziale della prima occorrenza di un modello in una determinata espressione su tutti i tipi di dati di carattere e testo validi. Restituisce zero se il modello non viene trovato. Durante la conversione da SQL Server ad Amazon RDS for PostgreSQL, AWS SCT `<pattern character><expression character varying>` sostituisce il codice dell'applicazione che utilizza `PATINDEX` con `aws_sqlserver_ext.patindex (,)`.

- In SQL Server, un tipo di tabella definita dall'utente è un tipo che rappresenta la definizione di una struttura di tabella. Utilizza un tipo di tabella definito dall'utente per dichiarare i parametri del valore di tabella per le stored procedure o le funzioni. È inoltre possibile utilizzare un tipo di tabella definito dall'utente per dichiarare le variabili di tabella che si desidera utilizzare in un batch o nel corpo di una procedura o funzione memorizzata. AWS SCT ha emulato questo tipo in PostgreSQL creando una tabella temporanea.

Durante la conversione da SQL Server a PostgreSQL, AWS SCT converte gli oggetti di sistema SQL Server in oggetti riconoscibili in PostgreSQL. La tabella seguente mostra il modo in cui vengono convertiti gli oggetti di sistema.

Casi d'uso di MS SQL Server	Sostituzione PostgreSQL
SYS.SCHEMAS	AWS_SQLSERVER_EXT.SYS_SCHEMAS
SYS.TABLES	AWS_SQLSERVER_EXT.SYS_TABLES
SYS.VIEWS	AWS_SQLSERVER_EXT.SYS_VIEWS
SYS.ALL_VIEWS	AWS_SQLSERVER_EXT.SYS_ALL_VIEWS
SYS.TYPES	AWS_SQLSERVER_EXT.SYS_TYPES
SYS.COLUMNS	AWS_SQLSERVER_EXT.SYS_COLUMNS
SYS.ALL_COLUMNS	AWS_SQLSERVER_EXT.SYS_ALL_COLUMNS
SYS.FOREIGN_KEYS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEYS
SYS.SYSFOREIGNKEYS	AWS_SQLSERVER_EXT.SYS_SYSFOREIGNKEYS
SYS.FOREIGN_KEY_COLUMNS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEY_COLUMNS
SYS.KEY_CONSTRAINTS	AWS_SQLSERVER_EXT.SYS_KEY_CONSTRAINTS
SYS.IDENTITY_COLUMNS	AWS_SQLSERVER_EXT.SYS_IDENTITY_COLUMNS

Casi d'uso di MS SQL Server	Sostituzione PostgreSQL
SYS.PROCEDURES	AWS_SQLSERVER_EXT.SYS_PROCEDURES
SYS.INDEXES	AWS_SQLSERVER_EXT.SYS_INDEXES
SYS.SYSINDEXES	AWS_SQLSERVER_EXT.SYS_SYSINDEXES
SYS.OBJECTS	AWS_SQLSERVER_EXT.SYS_OBJECTS
SYS.ALL_OBJECTS	AWS_SQLSERVER_EXT.SYS_ALL_OBJECTS
SYS.SYSOBJECTS	AWS_SQLSERVER_EXT.SYS_SYSOBJECTS
SYS.SQL_MODULES	AWS_SQLSERVER_EXT.SYS_SQL_MODULES
SYS.DATABASES	AWS_SQLSERVER_EXT.SYS_DATABASES
INFORMATION_SCHEMA .SCHEMATA	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_SCHEMATA
INFORMATION_SCHEMA .VIEWS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_VIEWS
INFORMATION_SCHEMA .TABLES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLES
INFORMATION_SCHEMA .COLUMNS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_COLUMNS
INFORMATION_SCHEMA .CHECK_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CHECK_CONSTRAINTS
INFORMATION_SCHEMA .REFERENTIAL_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_REFERENTIAL_CONSTRAINTS
INFORMATION_SCHEMA .TABLE_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLE_CONSTRAINTS

Casi d'uso di MS SQL Server	Sostituzione PostgreSQL
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
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

Utilizzare unAWS SCTpacchetto di estensioni per emulare SQL Server Agent in PostgreSQL

SQL Server Agent è un servizio di Microsoft Windows che esegue processi di SQL Server. SQL Server Agent esegue i job in base a una pianificazione, in risposta a un evento specifico o da dall'istanza. Per ulteriori informazioni relative a SQL Server Agent, vedere [Documentazione tecnica Microsoft](#).

PostgreSQL non ha un equivalente per SQL Server Agent. Per emulare le funzionalità di SQL Server Agent,AWS SCTcrea un pacchetto di estensione. Questo pacchetto di estensione utilizzaAWS Lambdae Amazon CloudWatch.AWS Lambdaimplementa l'interfaccia utilizzata per gestire le pianificazioni ed eseguire i lavori. Amazon CloudWatch mantiene le regole di pianificazione.

AWS Lambdae Amazon CloudWatch usa un parametro JSON per interagire. Questo parametro JSON ha la seguente struttura.

```
{
  "mode": mode,
```

```

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

```

Nell'esempio precedente, *mode* è il tipo di attività e *list of parameters* è un insieme di parametri che dipendono dal tipo di attività. Inoltre, *procedure name* è il nome della procedura che viene eseguita dopo il completamento dell'attività.

AWS SCT utilizza una funzione Lambda per controllare ed eseguire i job. La CloudWatch la regola avvia l'esecuzione del processo e fornisce le informazioni necessarie per avviare il processo. Quando CloudWatch rule trigger, avvia la funzione Lambda utilizzando i parametri della regola.

Per creare un job semplice che richiama una procedura, utilizzare il formato seguente.

```

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

```

Per creare un lavoro con diverse fasi, utilizzare il formato seguente.

```

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

Per emulare il comportamento di SQL Server Agent in PostgreSQL, AWS SCT il pacchetto di estensione crea anche le seguenti tabelle e procedure.

Tabelle che emulano SQL Server Agent in PostgreSQL

Per emulare SQL Server Agent, il pacchetto di estensione utilizza le seguenti tabelle:

sysjobs

Memorizza le informazioni relative ai lavori.

sysjobsteps

Memorizza le informazioni relative ai passaggi del processo.

pianificazioni di sistema

Memorizza le informazioni relative ai piani di lavoro.

sysjobschedules

Memorizza le informazioni sulla pianificazione per i singoli lavori.

sysjobhistory

Memorizza le informazioni relative ai processi pianificati.

Procedure che emulano SQL Server Agent in PostgreSQL

Per emulare SQL Server Agent, il pacchetto di estensione utilizza le seguenti procedure:

sp_add_job

Aggiunge un nuovo lavoro.

sp_add_jobstep

Aggiunge un passaggio a un lavoro.

sp_add_schedule

Crea una nuova regola di pianificazione in Amazon CloudWatch. È possibile utilizzare questa pianificazione con un numero qualsiasi di lavori.

sp_attach_schedule

Imposta una pianificazione per il lavoro selezionato.

sp_add_jobschedule

Crea una regola di pianificazione per un lavoro in Amazon CloudWatch e imposta l'obiettivo di questa regola.

sp_update_job

Aggiorna gli attributi del lavoro creato in precedenza.

sp_update_jobstep

Aggiorna gli attributi della fase di un lavoro.

sp_update_schedule

Aggiorna gli attributi di una regola di pianificazione in Amazon CloudWatch.

sp_update_jobschedule

Aggiorna gli attributi della pianificazione relativa al lavoro specificato.

sp_delete_job

Elimina un lavoro.

sp_delete_jobstep

Elimina una fase di lavoro da un lavoro.

sp_delete_schedule

Elimina una pianificazione.

sp_delete_jobschedule

Elimina la regola di pianificazione per il lavoro specificato da Amazon CloudWatch.

sp_detach_schedule

Rimuove un'associazione tra una pianificazione e un lavoro.

get_jobs, update_job

Procedure interne che interagiscono con AWS Elastic Beanstalk.

sp_verify_job_date, sp_verify_job_time, sp_verify_job, sp_verify_jobstep, sp_verify_schedule, sp_verify_job_identifiers, sp_verify_schedule_identifiers, sp_verify_schedule_identifiers

Procedure interne per il controllo delle impostazioni.

Sintassi per le procedure che emulano SQL Server Agent in PostgreSQL

Il `llaws_sqlserver_ext.sp_add_job` è una procedura nel pacchetto di estensione emula `ilmsdb.dbo.sp_add_job` di SQL Server Agent. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sp_add_jobstepla procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_add_jobstepprocedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

llaws_sqlserver_ext.sp_add_schedulela procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_add_scheduleprocedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

llaws_sqlserver_ext.sp_attach_schedule la procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_attach_schedule procedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sp_add_jobschedule la procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_add_jobschedule procedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sp_delete_jobla procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_delete_jobprocedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sp_delete_jobstepla procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_delete_jobstepprocedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica Microsoft](#).

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

Laaws_sqlserver_ext.sp_delete_jobschedulela procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_delete_jobscheduleprocedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sp_delete_schedule la procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_delete_schedule procedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sp_detach_schedule la procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_detach_schedule procedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sp_update_job la procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_update_job procedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sp_update_jobschedulela procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_update_jobscheduleprocedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sp_update_jobstepla procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_update_jobstepprocedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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

```

Laaws_sqlserver_ext.sp_update_schedulela procedura nel pacchetto di estensione emula ilmsdb.dbo.sp_update_scheduleprocedura. Per ulteriori informazioni sulla procedura di SQL Server Agent, vedere [Documentazione tecnica 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

```

Esempi di utilizzo di procedure che emulano SQL Server Agent in PostgreSQL

Per aggiungere un nuovo lavoro, utilizzare ilaws_sqlserver_ext.sp_add_jobprocedura come indicato di seguito.

```

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');
```

Per aggiungere una nuova fase di lavoro, utilizzare `aws_sqlserver_ext.sp_add_jobstep` procedura come indicato di seguito.

```
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');
```

Per aggiungere una pianificazione semplice, utilizzare `aws_sqlserver_ext.sp_add_schedule` procedura come indicato di seguito.

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

Per impostare una pianificazione per un lavoro, utilizzare `aws_sqlserver_ext.sp_attach_schedule` procedura come indicato di seguito.

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

Per creare una pianificazione per un lavoro, utilizzare `aws_sqlserver_ext.sp_add_jobschedule` procedura come indicato di seguito.

```
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);
```

Esempi di casi d'uso per l'emulazione di SQL Server Agent in PostgreSQL

Se il codice del database sorgente utilizza SQL Server Agent per eseguire i job, è possibile utilizzare il pacchetto di estensione SQL Server to PostgreSQL per AWS SCT per convertire questo codice in PostgreSQL. Il pacchetto di estensione utilizza AWS Lambda funzioni per emulare il comportamento di SQL Server Agent.

Puoi creare un nuovo AWS Lambda funzione o registra una funzione esistente.

Per creare una nuova funzione AWS Lambda

1. In AWS SCT, nell'albero database di destinazione, apri il menu di scelta rapida da dall'istanza database di destinazione, scegli **Applica il pacchetto di estensione per** e quindi **scegliere PostgreSQL**.

Viene visualizzata la procedura guidata del pacchetto di estensione.

2. Sul Servizio di emulazione di SQL Server Agent **scheda**, effettuare le operazioni di seguito.
 - Scegli **Creare un AWS Lambda funzione**.
 - Per **Accesso al database**, immettere il nome dell'utente del database di destinazione.
 - Per **Password del database**, immettere la password per il nome utente da dall'istanza precedente.
 - Per **cartella della libreria Python**, inserisci il percorso della cartella della libreria Python.
 - Scegli **Crea AWS Lambda funzione** e quindi **scegliere Avanti**.

Per registrare un AWS Lambda funzione che hai implementato in precedenza

- Esegui lo script seguente nel database di destinazione.

```
SELECT
```



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

Nell'esempio precedente, **ARN** è l'Amazon Resource Name (ARN) del modulo distribuito AWS Lambda funzione.

L'esempio seguente crea un'operazione semplice da eseguire da dall'istanza. Ogni cinque minuti, questa attività esegue l'operazione creata in precedenza `job_example` funzione. Questa funzione inserisce i record nel `job_example_table` tavolo.

Per creare questa semplice operazione

1. Creare i lavori utilizzando `aws_sqlserver_ext.sp_add_job` funzione come indicato di seguito.

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

2. Creare una fase di lavoro utilizzando il `aws_sqlserver_ext.sp_add_jobstep` funzione come indicato di seguito.

```
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;');
```

La fase di lavoro specifica cosa fa la funzione.

3. Crea uno scheduler per il lavoro utilizzando il `aws_sqlserver_ext.sp_add_jobschedule` funzione come indicato di seguito.

```
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 */);
```

La fase di lavoro specifica cosa fa la funzione.

Per eliminare questo lavoro, utilizzare `ilaws_sqlserver_ext.sp_delete_job` funzione come indicato di seguito.

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

Utilizzare unAWS SCTpacchetto di estensione per emulare SQL Server Database Mail in PostgreSQL

Puoi utilizzare SQL Server Database Mail per inviare messaggi di posta elettronica agli utenti da dall'istanza database di SQL Server. dall'istanza database di SQL Server dall'istanza database di SQL Server. Questi messaggi di posta elettronica possono contenere risultati di query o includere file provenienti da qualsiasi risorsa della rete. Per ulteriori informazioni relative alla posta del database di SQL Server, vedere [Documentazione tecnica Microsoft](#).

PostgreSQL non ha un equivalente per SQL Server Database Mail. Per emulare le funzionalità di SQL Server Database Mail,AWS SCTcrea un pacchetto di estensione. Questo pacchetto di estensione utilizzaAWS Lambdae Amazon Simple Email Service (Amazon SES).AWS Lambdafornece agli utenti un'interfaccia per interagire con il servizio di invio e-mail di Amazon SES. Per configurare questa interazione, aggiungi Amazon Resource Name (ARN) da Amazon Resource Name (ARN) da Lambda.

Per un nuovo account di posta elettronica, utilizzare il comando seguente.

```
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;
```

Per aggiungere l'ARN della funzione Lambda all'account e-mail esistente, utilizzare il comando seguente.

```
do  
$$  
begin  
PERFORM sysmail_update_account_sp (  
    par_account_name := 'existind_account_name',  
    par_mailserver_type := 'AWSLAMBDA'  
    par_mailserver_name := 'ARN'  
);  
end;  
$$ language plpgsql;
```

Negli esempi precedenti, **ARN** è l'ARN della funzione Lambda.

Per emulare il comportamento di SQL Server Database Mail in PostgreSQL, AWS SCT il pacchetto di estensione utilizza le tabelle, le viste e le procedure seguenti.

Tabelle che emulano la posta del database SQL Server in PostgreSQL

Per emulare SQL Server Database Mail, il pacchetto di estensione utilizza le seguenti tabelle:

`sysmail_account`

Memorizza le informazioni relative agli account di posta elettronica.

`sysmail_profile`

Memorizza le informazioni relative ai profili utente.

`sysmail_server`

Memorizza le informazioni relative ai server di posta elettronica.

`sysmail_mailitems`

Memorizza l'elenco dei messaggi di posta elettronica.

sysmail_attachments

Contiene una riga per ogni allegato di posta elettronica.

sysmail_log

Memorizza le informazioni di servizio relative all'invio di messaggi di posta elettronica.

sysmail_profileaccount

Memorizza le informazioni relative ai profili utente e agli account di posta elettronica.

Visualizzazioni che emulano la posta del database SQL Server in PostgreSQL

Per emulare SQL Server Database Mail, AWS SCT crea le seguenti viste nel database PostgreSQL per garantire la compatibilità. Il pacchetto di estensione non le utilizza, ma il codice convertito può interrogare queste viste.

sysmail_allitems

Include un elenco di tutti i messaggi di posta elettronica.

sysmail_faileditems

Include un elenco di e-mail che non è stato possibile inviare.

sysmail_sentitems

Include un elenco di e-mail inviate.

sysmail_unsentitems

Include un elenco di e-mail che non sono ancora state inviate.

sysmail_mailattachments

Include un elenco di file allegati.

Procedure che emulano la posta del database SQL Server in PostgreSQL

Per emulare SQL Server Database Mail, il pacchetto di estensione utilizza le seguenti procedure:

sp_send_dbmail

Invia un'e-mail ai destinatari specificati.

sysmail_add_profile_sp

Crea un nuovo profilo utente.

sysmail_add_account_sp

Crea un nuovo account e-mail che memorizza informazioni come le credenziali SMTP (Simple Mail Transfer Protocol) e così via.

sysmail_add_profileaccount_sp

Aggiunge un account e-mail al profilo utente specificato.

sysmail_update_profile_sp

Modifica gli attributi del profilo utente come descrizione, nome e così via.

sysmail_update_account_sp

Modifica le informazioni nell'account di posta elettronica esistente.

sysmail_update_profileaccount_sp

Aggiorna le informazioni dell'account e-mail nel profilo utente specificato.

sysmail_delete_profileaccount_sp

Rimuove un account e-mail dal profilo utente specificato.

sysmail_delete_account_sp

Elimina l'account di posta elettronica.

sysmail_delete_profile_sp

Elimina il profilo utente.

sysmail_delete_mailitems_sp

Elimina i messaggi di posta elettronica dalle tabelle interne.

sysmail_help_profile_sp

Visualizza informazioni sul profilo utente.

sysmail_help_account_sp

Visualizza informazioni sull'account di posta elettronica.

sysmail_help_profileaccount_sp

Visualizza informazioni sugli account di posta elettronica associati al profilo utente.

sysmail_dbmail_json

Una procedura interna che genera richieste JSON perAWS Lambdafunzioni.

sysmail_verify_profile_sp, sysmail_verify_account_sp, sysmail_verify_addressparams_sp

Procedure interne per il controllo delle impostazioni.

sp_get_dbmail, sp_set_dbmail, sysmail_dbmail_xml

Procedure interne obsolete.

Sintassi per le procedure che emulano SQL Server Database Mail in PostgreSQL

Il `llaws_sqlserver_ext.sp_send_dbmail` la procedura nel pacchetto di estensione emula il `ilmsdb.dbo.sp_send_dbmail` procedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sysmail_delete_mailitems_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_delete_mailitems_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica Microsoft](#).

```
par_sent_before timestamp = NULL::timestamp without time zone,  
par_sent_status varchar = NULL::character varying,  
out returncode integer
```

Laaws_sqlserver_ext.sysmail_add_profile_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_add_profile_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica Microsoft](#).

```
par_profile_name varchar,  
par_description varchar = NULL::character varying,  
out par_profile_id integer,  
out returncode integer
```

Laaws_sqlserver_ext.sysmail_add_account_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_add_account_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sysmail_add_profileaccount_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_add_profileaccount_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sysmail_help_profile_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_help_profile_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
out returncode integer
```

Laaws_sqlserver_ext.sysmail_update_profile_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_update_profile_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_description varchar = NULL::character varying,  
out returncode integer
```

Laaws_sqlserver_ext.sysmail_delete_profile_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_delete_profile_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_force_delete smallint = 1,  
out returncode integer
```


Laaws_sqlserver_ext.sysmail_help_account_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_help_account_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

Laaws_sqlserver_ext.sysmail_update_account_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_update_account_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sysmail_delete_account_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_delete_account_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

Laaws_sqlserver_ext.sysmail_help_profileaccount_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_help_profileaccount_spprocedura. Per ulteriori

informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sysmail_update_profileaccount_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_update_profileaccount_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica 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
```

Laaws_sqlserver_ext.sysmail_delete_profileaccount_spla procedura nel pacchetto di estensione emula ilmsdb.dbo.sysmail_delete_profileaccount_spprocedura. Per ulteriori informazioni sulla procedura di posta elettronica database di SQL Server, vedere [Documentazione tecnica 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
```

Esempi di utilizzo di procedure che emulano SQL Server Database Mail in PostgreSQL

Per inviare un messaggio di posta elettronica, utilizzare ilaws_sqlserver_ext.sp_send_dbmailprocedura come indicato di seguito.

```
PERFORM sp_send_dbmail (  
    par_profile_name := 'Administrator',  
    par_recipients := 'hello@rusgl.info',  
    par_subject := 'Automated Success Message',
```

```
    par_body := 'The stored procedure finished'
);
```

L'esempio seguente mostra come inviare un messaggio di posta elettronica con risultati di query.

```
PERFORM sp_send_dbmail (
    par_profile_name := 'Administrator',
    par_recipients := 'hello@rusgl.info',
    par_subject := 'Account with id = 1',
    par_query := 'SELECT COUNT(*)FROM Account WHERE id = 1'
);
```

L'esempio seguente mostra come inviare un'e-mail con codice 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'
);
```

Per eliminare i messaggi di posta elettronica, utilizzare `ilaws_sqlserver_ext.sysmail_delete_mailitems_spprocedura` come indicato di seguito.

```
DECLARE var_GETDATE datetime;
SET var_GETDATE = NOW();
PERFORM sysmail_delete_mailitems_sp (
    par_sent_before := var_GETDATE
);
```

L'esempio seguente mostra come eliminare le e-mail più vecchie.

```
PERFORM sysmail_delete_mailitems_sp (
```

```
    par_sent_before := '31.12.2015'  
);
```

L'esempio seguente mostra come eliminare tutte le email che non possono essere inviate.

```
PERFORM sysmail_delete_mailitems_sp (  
    par_sent_status := 'failed'  
);
```

Per creare un nuovo profilo utente, utilizzare `ilaws_sqlserver_ext.sysmail_add_profile_sprocedura` come indicato di seguito.

```
PERFORM sysmail_add_profile_sp (  
    profile_name := 'Administrator',  
    par_description := 'administrative mail'  
);
```

L'esempio seguente mostra come creare un nuovo profilo e salvare l'identificatore di profilo da dall'istanza variabile.

```
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;
```

Per creare un nuovo account di posta elettronica, utilizzare `ilaws_sqlserver_ext.sysmail_add_account_sprocedura` come indicato di seguito.

```
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'  
);
```

Per aggiungere un account e-mail al profilo utente, utilizzare `ilaws_sqlserver_ext.sysmail_add_profileaccount_spprocedura` come indicato di seguito.

```
PERFORM sysmail_add_profileaccount_sp (  
    par_account_name := 'Administrator',  
    par_account_name := 'Audit Account',  
    par_sequence_number := 1  
);
```

Esempi di casi d'uso per l'emulazione di SQL Server Database Mail in PostgreSQL

Se il codice del database sorgente utilizza SQL Server Database Mail per inviare messaggi di posta elettronica, è possibile utilizzare AWS SCT pacchetto di estensione per convertire questo codice in PostgreSQL.

Per inviare un'e-mail dal database PostgreSQL

1. Crea e configura iAWS Lambda funzione.
2. Applica ilAWS SCT pacchetto di estensione.
3. Creare un profilo utente utilizzando il `sysmail_add_profile_sp` funzione come indicato di seguito.
4. Creare un account di posta elettronica utilizzando il `sysmail_add_account_sp` funzione come indicato di seguito.
5. Aggiungi questo account e-mail al tuo profilo utente utilizzando il `sysmail_add_profileaccount_sp` funzione come indicato di seguito.

```
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. Inviare i messaggi di posta elettronica `sp_send_dbmail` funzione come indicato di seguito.

```

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;

```

Per visualizzare le informazioni su tutti i profili utente, usare `sysmail_help_profile_sp` procedura come indicato di seguito.

```
SELECT FROM aws_sqlserver_ext.sysmail_help_profile_sp();
```

L'esempio seguente visualizza le informazioni da dall'istanza utente.

```

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');

```

Per visualizzare le informazioni su tutti gli account di posta elettronica, utilizza `sysmail_help_account_sp` come indicato di seguito.

```
select from aws_sqlserver_ext.sysmail_help_account_sp();
```

L'esempio seguente visualizza le informazioni da dall'account di posta elettronica specifico.

```
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');
```

Per visualizzare le informazioni su tutti gli account di posta elettronica associati ai profili utente, utilizza `sysmail_help_profileaccount_sp` come indicato di seguito.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp();
```

L'esempio seguente filtra i record in base all'identificatore, al nome del profilo o al nome dell'account.

```
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');
```

Per modificare il nome o la descrizione del profilo utente, utilizza `sysmail_update_profile_sp` come illustrato di seguito.

```
select aws_sqlserver_ext.sysmail_update_profile_sp(
par_profile_id := 2,
par_profile_name := 'New profile name'
);
```

Per modificare le impostazioni dell'account e-mail, utilizza `sysmail_update_account_sp` come illustrato di seguito.

```
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'  
);
```

Conversione di SQL Server in Amazon RDS per SQL Server

Alcuni aspetti da considerare durante la migrazione dello schema e del codice SQL Server ad Amazon RDS per SQL Server:

- AWS SCT può convertire SQL Server Agent per fornire pianificazioni, avvisi e processi su un'istanza DB di Amazon RDS for SQL Server. Dopo la conversione, è possibile usare un'istanza database Amazon RDS for SQL Server con SQL Server Reporting Service (SSRS), SQL Server Analysis Services (SSAS) e SQL Server Integration Services (SSIS).
- Amazon RDS non supporta attualmente SQL Server Service Broker o endpoint T-SQL aggiuntivi che richiedono l'esecuzione del comando CREATE ENDPOINT.
- Amazon RDS dispone di supporto limitato per i server collegati. Durante la conversione del codice applicativo SQL Server che utilizza server collegati, AWS SCT converte il codice dell'applicazione. Tuttavia, assicurarsi di esaminare il comportamento degli oggetti che utilizzano server collegati prima di eseguire il codice convertito.
- Si utilizza Sempre acceso.
- La AWS SCT il rapporto di valutazione fornisce le metriche del server per la conversione. Questi parametri sull'istanza SQL Server includono quanto segue:
 - Viene utilizzato il mirroring dei dati.
 - SQL Server Log Shipping è configurato.
 - Il cluster di failover viene utilizzato.
 - Database Mail è configurato.
 - Full Text Search Service viene utilizzato. Amazon RDS per SQL Server ha una ricerca full-text limitata e non supporta la ricerca semantica.
 - Data Quality Service (DQS) è installato. Amazon RDS non supporta DQS quindi consigliamo di installare SQL Server su un'istanza Amazon EC2.

Privilegi per RDS for SQL Server come destinazione

Per eseguire la migrazione a RDS per SQL Server, crea un utente del database e quindi concedi i privilegi richiesti per ogni database. È possibile utilizzare il seguente esempio di codice.


```
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;
```

Nell'esempio precedente, sostituire *nome_utente* con il nome del tuo utente. Quindi, sostituisci *db_name* con il nome del database di destinazione. Infine, sostituisci *your_password* con una password sicura.

Fonti di data warehouse per AWS Schema Conversion Tool

AWS SCT può convertire gli schemi per i seguenti data warehouse di origine in una destinazione supportata. Per informazioni su autorizzazioni, connessioni e cosa è AWS SCT possibile convertire per l'utilizzo con il database o il data warehouse di destinazione, vedere i dettagli di seguito.

Argomenti

- [Utilizzo di Amazon Redshift come fonte per AWS SCT](#)
- [Utilizzo di Azure Synapse Analytics come fonte per AWS SCT](#)
- [Utilizzo di BigQuery come origine per AWS SCT](#)
- [Utilizzo del database Greenplum come fonte per AWS SCT](#)
- [Usare Netezza come fonte per AWS SCT](#)
- [Utilizzo di Oracle Data Warehouse come fonte per AWS SCT](#)
- [Usare Snowflake come fonte per AWS SCT](#)

- [Utilizzo di Microsoft SQL Server Data Warehouse come fonte per AWS SCT](#)
- [Utilizzo di Teradata come fonte per AWS SCT](#)
- [Usare Vertica come fonte per AWS SCT](#)

Utilizzo di Amazon Redshift come fonte per AWS SCT

Puoi utilizzarlo AWS SCT per ottimizzare il tuo cluster Amazon Redshift. AWS SCT fornisce consigli sulla selezione delle chiavi di distribuzione e ordinamento per il tuo cluster Amazon Redshift. Puoi considerare il progetto di ottimizzazione di Amazon Redshift come un AWS SCT progetto con origine e destinazione che puntano ai diversi cluster Amazon Redshift.

Privilegi per Amazon Redshift come database di origine

Per utilizzare Amazon Redshift come fonte sono necessari i seguenti privilegi:

- USAGE ON SCHEMA *<schema_name>*
- SELECT ON ALL TABLES IN SCHEMA *<schema_name>*
- SELECT ON PG_CATALOG.PG_STATISTIC
- SELECT ON SVV_TABLE_INFO
- SELECT ON TABLE STV_BLOCKLIST
- SELECT ON TABLE STV_TBL_PERM
- SELEZIONA SYS_SERVERLESS_USAGE
- SELEZIONA SU PG_DATABASE_INFO
- SELEZIONA SU PG_STATISTIC

Negli esempi precedenti, sostituire il *<schema_name>* segnaposto con il nome dello schema di origine.

Per i privilegi richiesti per Amazon Redshift come target, consulta. [Autorizzazioni per Amazon Redshift come obiettivo](#)

Connessione ad Amazon Redshift come fonte

Utilizza la seguente procedura per connetterti al tuo database di origine Amazon Redshift con. AWS Schema Conversion Tool

Per connettersi a un database di origine Amazon Redshift

1. NelAWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Amazon Redshift, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCTvisualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per AWSSecret, scegli il nome del segreto.
 2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire le informazioni di connessione per il database di origine di Amazon Redshift, utilizza le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Inserisci il nome DNS (Domain Name System) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
Database	Immettere Il nome del database Amazon Redshift.
User name (Nome utente) e Password (Password)	Inserisci le credenziali del database per connetterti al server del database di origine. AWS SCTutilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto

Parametro	Operazione
	AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.
Use SSL (Usa SSL)	<p>Scegli questa opzione per utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none">• Verifica il certificato del server: seleziona questa opzione per verificare il certificato del server utilizzando un archivio attendibile.• Trust store: la posizione di un trust store contenente certificati. Affinché questa posizione appaia qui, assicurati di aggiungerla nelle Impostazioni globali. <p>Per ulteriori informazioni sul supporto SSL per Amazon Redshift, consulta Configurare le opzioni di sicurezza per le connessioni.</p>
Memorizza la password	AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza dover inserire la password.
Percorso del driver Redshift	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

5. Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
6. Scegli Connetti per connetterti al tuo database di origine.

Impostazioni di ottimizzazione di Amazon Redshift

Per modificare le impostazioni di ottimizzazione di Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Amazon Redshift, quindi scegli Amazon Redshift — Amazon Redshift. AWS SCT visualizza tutte le impostazioni disponibili per l'ottimizzazione di Amazon Redshift.

Le impostazioni di ottimizzazione di Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungi commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per impostare il numero massimo di tabelle che AWS SCT possono essere applicate al cluster Amazon Redshift di destinazione.

Per il numero massimo di tabelle per il cluster Amazon Redshift di destinazione, scegli il numero di tabelle che AWS SCT possono essere applicate al tuo cluster Amazon Redshift.

Amazon Redshift prevede quote che limitano le tabelle di utilizzo per diversi tipi di nodi del cluster. Se scegli Auto, AWS SCT determina il numero di tabelle da applicare al cluster Amazon Redshift di destinazione in base al tipo di nodo. Facoltativamente, scegli il valore manualmente. Per ulteriori informazioni, consulta [Quote e limiti in Amazon Redshift](#) nella Guida alla gestione di Amazon Redshift.

AWS SCT converte tutte le tabelle di origine, anche se il numero di tabelle è superiore a quello che può memorizzare il cluster Amazon Redshift. AWS SCT memorizza il codice convertito nel progetto e non lo applica al database di destinazione. Se raggiungi la quota del cluster Amazon Redshift per le tabelle quando applichi il codice convertito, viene visualizzato un messaggio di avviso. Inoltre, AWS SCT applica le tabelle al cluster Amazon Redshift di destinazione fino a quando il numero di tabelle raggiunge il limite.

- Per scegliere la strategia di migrazione.

AWS consiglia di utilizzare cluster diversi come origine e destinazione per il progetto di ottimizzazione. Prima di iniziare il processo di ottimizzazione di Amazon Redshift, crei una copia

del cluster Amazon Redshift di origine. Puoi includere i tuoi dati di origine in questa copia o creare un cluster vuoto.

Per la strategia di migrazione, scegli Migrazione verso una copia per includere i dati dal cluster di origine nel cluster di destinazione.

Per la strategia di migrazione, scegli Migrazione a tabula rasa per esaminare i suggerimenti di ottimizzazione. Dopo aver accettato questi suggerimenti, esegui la migrazione dei dati di origine al cluster di destinazione.

- Per applicare la compressione alle colonne della tabella Amazon Redshift. A tale scopo, seleziona Usa la codifica di compressione.

AWS SCT assegna automaticamente la codifica di compressione alle colonne utilizzando l'algoritmo Amazon Redshift predefinito. Per ulteriori informazioni, consulta [Codifiche di compressione](#) nella Amazon Redshift Database Developer Guide.

Per impostazione predefinita, Amazon Redshift non applica la compressione alle colonne definite come chiavi di ordinamento e distribuzione. È possibile modificare questo comportamento e applicare la compressione a queste colonne. A tale scopo, seleziona Usa la codifica di compressione per le colonne KEY. È possibile selezionare questa opzione solo quando è stata selezionata l'opzione Usa codifica di compressione.

- Per lavorare con l'ottimizzazione automatica delle tabelle.

L'ottimizzazione automatica delle tabelle è un processo di ottimizzazione automatica in Amazon Redshift che ottimizza automaticamente la progettazione delle tabelle. Per ulteriori informazioni, consulta [Working with Automatic Tabelle Optimization](#) nella Amazon Redshift Database Developer Guide.

Da utilizzare solo per l'ottimizzazione automatica delle tabelle, scegli Strategie di ottimizzazione nel riquadro a sinistra. Quindi seleziona Usa l'ottimizzazione automatica delle tabelle di Amazon Redshift e scegli Nessuno per la strategia di selezione iniziale delle chiavi.

- Per scegliere le chiavi di ordinamento e distribuzione utilizzando la tua strategia.

Puoi scegliere le chiavi di ordinamento e distribuzione utilizzando i metadati di Amazon Redshift, le informazioni statistiche o entrambe queste opzioni. Per la strategia di selezione iniziale delle chiavi nella scheda Strategie di ottimizzazione, scegli una delle seguenti opzioni:

- Usa i metadati, ignora le informazioni statistiche
- Ignora i metadati, utilizza le informazioni statistiche

- Usa metadati e informazioni statistiche

A seconda dell'opzione scelta, puoi selezionare strategie di ottimizzazione. Quindi, per ogni strategia, inserisci il valore (0-100). Questi valori definiscono il peso di ogni strategia. Utilizzando questi valori di peso, AWS SCT definisce in che modo ciascuna regola influisce sulla scelta delle chiavi di distribuzione e ordinamento. I valori predefiniti si basano sulle best practice di AWS migrazione.

È possibile definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle di piccole dimensioni. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserite il numero minimo e massimo di righe in una tabella per definirla come una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

- Per configurare i dettagli della strategia.

Oltre a definire il peso di ogni strategia di ottimizzazione, puoi configurare le impostazioni di ottimizzazione. Per farlo, scegli Ottimizzazione delle conversioni.

- In Limite colonne chiave di ordinamento, immettete il numero massimo di colonne nella chiave di ordinamento.
- In Valore di soglia inclinato, inserisci la percentuale (0-100) di un valore inclinato per una colonna. AWS SCT esclude le colonne con un valore di inclinazione superiore alla soglia dall'elenco dei candidati per la chiave di distribuzione. AWS SCT definisce il valore distorto per una colonna come il rapporto percentuale tra il numero di occorrenze del valore più comune e il numero totale di record.
- Per le prime N query della tabella della cronologia delle query, inserisci il numero (1—100) delle query utilizzate più di frequente da analizzare.
- In Seleziona utente delle statistiche, scegli l'utente del database di cui desideri analizzare le statistiche delle query.

Utilizzo di Azure Synapse Analytics come fonte per AWS SCT

Puoi usarlo AWS SCT per convertire schemi, oggetti di codice e codice applicativo da Azure Synapse Analytics ad Amazon Redshift.

Privilegi per Azure Synapse Analytics come database di origine

I seguenti privilegi sono necessari per utilizzare un data warehouse di Azure Synapse Analytics come origine:

- VIEW DEFINITION
- VIEW DATABASE STATE

Applica i privilegi per ogni database di cui stai convertendo lo schema.

Connessione ad Azure Synapse Analytics come fonte

Usa la procedura seguente per connetterti al tuo data warehouse di Azure Synapse Analytics con AWS Schema Conversion Tool

Per connettersi a un data warehouse di Azure Synapse Analytics come origine

1. Nel AWS Schema Conversion Tool, scegli **Aggiungi fonte**.
2. Scegli **Azure Synapse Analytics**, quindi scegli **Avanti**.

Viene visualizzata la finestra di dialogo **Aggiungi fonte**.

3. In **Nome connessione**, inserisci un nome per il tuo database. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per **AWS Secret**, scegli il nome del segreto.
 2. Scegliete **Compila** per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione per il data warehouse di Azure Synapse Analytics, usa le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Immetti il nome Domain Name Service (DNS) o l'indirizzo IP del server del database di origine.
Pool SQL	Inserisci il nome del pool SQL di Azure.
User name (Nome utente) e Password (Password)	<p>Inserisci le credenziali del database per connetterti al server del database di origine.</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p>
Use SSL (Usa SSL)	<p>Scegli questa opzione per utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none"> • Certificato server attendibile: scegli questa opzione per considerare attendibile il certificato del server. • Trust store: un trust store che hai configurato nelle Impostazioni globali.
Memorizza la password	<p>AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza immettere la password.</p>

5. Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
6. Scegli Connetti per connetterti al tuo database di origine.

Impostazioni di conversione da Azure Synapse Analytics ad Amazon Redshift

Per modificare le impostazioni di conversione da Azure Synapse Analytics ad Amazon Redshift, scegli Impostazioni inAWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Azure Synapse, quindi scegli Azure Synapse — Amazon Redshift. AWS SCTvisualizza tutte le impostazioni disponibili per la conversione da Azure Synapse Analytics ad Amazon Redshift.

Le impostazioni di conversione da Azure Synapse Analytics ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungi commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCTaggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per impostare il numero massimo di tabelle che AWS SCT possono essere applicate al cluster Amazon Redshift di destinazione.

Per Il numero massimo di tabelle per il cluster Amazon Redshift di destinazione, scegli il numero di tabelle che AWS SCT possono essere applicate al tuo cluster Amazon Redshift.

Amazon Redshift prevede quote che limitano le tabelle di utilizzo per diversi tipi di nodi del cluster. Se scegli Auto, AWS SCT determina il numero di tabelle da applicare al cluster Amazon Redshift di destinazione in base al tipo di nodo. Facoltativamente, scegli il valore manualmente. Per ulteriori informazioni, consulta [Quote e limiti in Amazon Redshift](#) nella Guida alla gestione di Amazon Redshift.

AWS SCTconverte tutte le tabelle di origine, anche se sono più di quelle che il tuo cluster Amazon Redshift può archiviare. AWS SCTmemorizza il codice convertito nel progetto e non lo applica al database di destinazione. Se raggiungi la quota del cluster Amazon Redshift per le tabelle quando applichi il codice convertito, viene AWS SCT visualizzato un messaggio di avviso. Inoltre, AWS SCT applica le tabelle al cluster Amazon Redshift di destinazione fino a quando il numero di tabelle raggiunge il limite.

- Per migrare le partizioni della tabella di origine in tabelle separate in Amazon Redshift. A tale scopo, seleziona Usa la vista UNION ALL e inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per una singola tabella di origine.

Amazon Redshift non supporta il partizionamento delle tabelle. Per emulare questo comportamento e velocizzare l'esecuzione delle query, AWS SCT puoi migrare ogni partizione della tabella di origine in una tabella separata in Amazon Redshift. Quindi, AWS SCT crea una vista che include i dati di tutte queste tabelle.

AWS SCT determina automaticamente il numero di partizioni nella tabella di origine. A seconda del tipo di partizionamento della tabella di origine, questo numero può superare la quota delle tabelle che puoi applicare al tuo cluster Amazon Redshift. Per evitare di raggiungere questa quota, inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per le partizioni di una singola tabella di origine. L'opzione predefinita è 368 tabelle, che rappresenta una partizione per 366 giorni all'anno e due tabelle per NO RANGE e partizioni. UNKNOWN

- Per applicare la compressione alle colonne della tabella Amazon Redshift. A tale scopo, seleziona Usa la codifica di compressione.

AWS SCT assegna automaticamente la codifica di compressione alle colonne utilizzando l'algoritmo Amazon Redshift predefinito. Per ulteriori informazioni, consulta [Codifiche di compressione](#) nella Amazon Redshift Database Developer Guide.

Per impostazione predefinita, Amazon Redshift non applica la compressione alle colonne definite come chiavi di ordinamento e distribuzione. È possibile modificare questo comportamento e applicare la compressione a queste colonne. A tale scopo, seleziona Usa la codifica di compressione per le colonne KEY. È possibile selezionare questa opzione solo quando si seleziona l'opzione Usa codifica di compressione.

Impostazioni di ottimizzazione della conversione da Azure Synapse Analytics ad Amazon Redshift

Per modificare le impostazioni di ottimizzazione della conversione da Azure Synapse Analytics ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Azure Synapse, quindi scegli Azure Synapse — Amazon Redshift. Nel riquadro a sinistra, scegli Strategie di ottimizzazione. AWS SCT visualizza le impostazioni di ottimizzazione della conversione per la conversione da Azure Synapse Analytics ad Amazon Redshift.

Le impostazioni di ottimizzazione della conversione da Azure Synapse Analytics ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per lavorare con l'ottimizzazione automatica delle tabelle. A tale scopo, seleziona Utilizza l'ottimizzazione automatica della tabella di Amazon Redshift.

L'ottimizzazione automatica delle tabelle è un processo di ottimizzazione automatica in Amazon Redshift che ottimizza automaticamente la progettazione delle tabelle. Per ulteriori informazioni, consulta [Working with Automatic Tabelle Optimization](#) nella Amazon Redshift Database Developer Guide.

Per fare affidamento solo sull'ottimizzazione automatica della tabella, scegli Nessuno per la strategia di selezione iniziale delle chiavi.

- Per scegliere le chiavi di ordinamento e distribuzione utilizzando la tua strategia.

Puoi scegliere le chiavi di ordinamento e distribuzione utilizzando i metadati di Amazon Redshift, le informazioni statistiche o entrambe queste opzioni. Per la strategia di selezione iniziale delle chiavi nella scheda Strategie di ottimizzazione, scegli una delle seguenti opzioni:

- Usa i metadati, ignora le informazioni statistiche
- Ignora i metadati, utilizza le informazioni statistiche
- Usa metadati e informazioni statistiche

A seconda dell'opzione scelta, puoi selezionare strategie di ottimizzazione. Quindi, per ogni strategia, inserisci il valore (0-100). Questi valori definiscono il peso di ogni strategia. Utilizzando questi valori di peso, AWS SCT definisce in che modo ciascuna regola influisce sulla scelta delle chiavi di distribuzione e ordinamento. I valori predefiniti si basano sulle best practice di AWS migrazione.

È possibile definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle di piccole dimensioni. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserite il numero minimo e massimo di righe in una tabella per definirla come una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

- Per configurare i dettagli della strategia.

Oltre a definire il peso di ogni strategia di ottimizzazione, puoi configurare le impostazioni di ottimizzazione. Per farlo, scegli Ottimizzazione delle conversioni.

- In Limite colonne chiave di ordinamento, immettete il numero massimo di colonne nella chiave di ordinamento.

- In Valore di soglia inclinato, inserisci la percentuale (0-100) di un valore inclinato per una colonna. AWS SCT esclude le colonne con un valore di inclinazione superiore alla soglia dall'elenco dei candidati per la chiave di distribuzione. AWS SCT definisce il valore distorto per una colonna come il rapporto percentuale tra il numero di occorrenze del valore più comune e il numero totale di record.
- Per le prime N query della tabella della cronologia delle query, inserisci il numero (1—100) delle query utilizzate più di frequente da analizzare.
- In Seleziona utente delle statistiche, scegli l'utente del database per il quale desideri analizzare le statistiche della query.

Inoltre, nella scheda Strategie di ottimizzazione, puoi definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle piccole. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserisci il numero minimo e massimo di righe in una tabella per considerarla una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

Utilizzo di BigQuery come origine per AWS SCT

Puoi utilizzarlo AWS SCT per convertire schemi, oggetti di codice e codice dell'applicazione BigQuery da Amazon Redshift.

Privilegi per BigQuery come fonte

Per utilizzare un BigQuery data warehouse come fonte in AWS SCT, crea un account di servizio. In Google Cloud, le applicazioni utilizzano gli account di servizio per effettuare chiamate API autorizzate. Gli account di servizio sono diversi dagli account utente. Per ulteriori informazioni, consulta [Account di servizio](#) nella documentazione di Google Cloud Identity and Access Management.

Assicurati di concedere i seguenti ruoli al tuo account di servizio:

- BigQuery Admin
- Storage Admin

Il BigQuery Admin ruolo fornisce le autorizzazioni per gestire tutte le risorse all'interno del progetto. AWS SCT utilizza questo ruolo per caricare i BigQuery metadati nel progetto di migrazione.

Il Storage Admin ruolo garantisce il pieno controllo degli oggetti e dei bucket di dati. Puoi trovare questo ruolo sotto Cloud Storage. AWS SCT utilizza questo ruolo per estrarre i dati da Amazon Redshift BigQuery e caricarli in Amazon Redshift.

Per creare un file chiave dell'account di servizio

1. Accedi alla console di gestione di Google Cloud all'[indirizzo https://console.cloud.google.com/](https://console.cloud.google.com/).
2. Nella pagina [BigQueryAPI](#), scegli Abilita. Ignora questo passaggio se vedi API Enabled.
3. Nella pagina [Account di servizio](#), scegli il tuo progetto, quindi scegli Crea account di servizio.
4. Nella pagina dei dettagli dell'account del servizio, inserisci un valore descrittivo per il nome dell'account del servizio. Scegli Crea e continua. Viene visualizzata la pagina Concedi a questo account di servizio l'accesso al progetto.
5. In Seleziona un ruolo, scegli BigQuery, quindi scegli BigQueryAmministratore.
6. Scegli Aggiungi un altro ruolo. Per Seleziona un ruolo, scegli Cloud Storage, quindi scegli Storage Admin.
7. Scegli Continua, quindi scegli Fine.
8. Nella pagina [Account di servizio](#), scegli l'account del servizio che hai creato.
9. Scegli Chiavi, quindi scegli Crea nuova chiave per Aggiungi chiave.
10. Scegli JSON, quindi scegli Crea. Scegli la cartella in cui salvare la tua chiave privata o seleziona la cartella predefinita per i download nel tuo browser.

Per estrarre dati da un BigQuery data warehouse, AWS SCT utilizza la cartella bucket di Google Cloud Storage. Crea questo bucket prima di iniziare la migrazione dei dati. Inserisci il percorso della cartella bucket di Google Cloud Storage nella finestra di dialogo Crea attività locale. Per ulteriori informazioni, consulta [Creazione, esecuzione e monitoraggio di qualsiasi AWS SCT attività](#).

Connessione a BigQuery come fonte

Usa la seguente procedura per connetterti al tuo BigQuery progetto sorgente con AWS Schema Conversion Tool.

Per connettersi a un data warehouse di BigQuery origine

1. Nel AWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli BigQuery, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome della connessione, inserisci un nome per il tuo BigQuery progetto. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.
4. In Percorso chiave, inserisci il percorso del file chiave dell'account del servizio. Per ulteriori informazioni sulla creazione di questo file, vedere [Privilegi per BigQuery come fonte](#).
5. Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo BigQuery progetto di origine.
6. Scegli Connetti per connetterti al tuo BigQuery progetto di origine.

Limitazioni BigQuery all'utilizzo come fonte per AWS SCT

Le seguenti limitazioni si applicano quando si utilizza BigQuery come fonte per AWS SCT:

- AWS SCT non supporta la conversione di sottoquery in funzioni analitiche.
- Non puoi utilizzarlo AWS SCT per convertire BigQuery `SELECT AS STRUCT` e `SELECT AS VALUE` dichiarare.
- AWS SCT non supporta la conversione dei seguenti tipi di funzioni:
 - Aggregato approssimativo
 - Bit
 - Debug
 - Query federata
 - Geografia
 - Hash
 - Matematiche
 - Rete
 - Aggregato statistico
 - UUID
- AWS SCT fornisce un supporto limitato per la conversione delle funzioni di stringa.
- AWS SCT non supporta la conversione degli `UNNEST` operatori.
- Non puoi convertire le operazioni di join correlate in AWS SCT.
- AWS SCT non supporta la conversione di `QUALIFY`, `WINDOWLIMIT`, e `OFFSET` clausole.
- Non puoi usare AWS SCT per convertire espressioni di tabella ricorsive comuni.

- AWS SCT non supporta la conversione di INSERT dichiarazioni con sottoquery all'interno VALUES di clausole.
- AWS SCT non supporta la conversione di UPDATE istruzioni per campi annidati e record ripetuti.
- Non puoi utilizzarlo AWS SCT per convertire STRUCT né per tipi di ARRAY dati.

BigQuery alle impostazioni di conversione di Amazon Redshift

BigQuery Per modificare le impostazioni di conversione di Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Google BigQuery, quindi scegli Google BigQuery — Amazon Redshift. AWS SCT visualizza tutte le impostazioni disponibili BigQuery per la conversione in Amazon Redshift.

BigQuery alle impostazioni di conversione di Amazon Redshift in AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungere commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per impostare il numero massimo di tabelle che AWS SCT possono essere applicate al cluster Amazon Redshift di destinazione.

Per Il numero massimo di tabelle per il cluster Amazon Redshift di destinazione, scegli il numero di tabelle che AWS SCT possono essere applicate al tuo cluster Amazon Redshift.

Amazon Redshift prevede quote che limitano le tabelle di utilizzo per diversi tipi di nodi del cluster. Se scegli Auto, AWS SCT determina il numero di tabelle da applicare al cluster Amazon Redshift di destinazione in base al tipo di nodo. Facoltativamente, scegli il valore manualmente. Per ulteriori informazioni, consulta [Quote e limiti in Amazon Redshift](#) nella Guida alla gestione di Amazon Redshift.

AWS SCT converte tutte le tabelle di origine, anche se sono più di quelle che il tuo cluster Amazon Redshift può archiviare. AWS SCT memorizza il codice convertito nel progetto e non lo applica al database di destinazione. Se raggiungi la quota del cluster Amazon Redshift per le tabelle quando

applichi il codice convertito, viene AWS SCT visualizzato un messaggio di avviso. Inoltre, AWS SCT applica le tabelle al cluster Amazon Redshift di destinazione fino a quando il numero di tabelle raggiunge il limite.

- Per applicare la compressione alle colonne della tabella Amazon Redshift. A tale scopo, seleziona Usa la codifica di compressione.

AWS SCT assegna automaticamente la codifica di compressione alle colonne utilizzando l'algoritmo Amazon Redshift predefinito. Per ulteriori informazioni, consulta [Codifiche di compressione](#) nella Amazon Redshift Database Developer Guide.

Per impostazione predefinita, Amazon Redshift non applica la compressione alle colonne definite come chiavi di ordinamento e distribuzione. È possibile modificare questo comportamento e applicare la compressione a queste colonne. A tale scopo, seleziona Usa la codifica di compressione per le colonne KEY. È possibile selezionare questa opzione solo quando si seleziona l'opzione Usa codifica di compressione.

BigQuery alle impostazioni di ottimizzazione della conversione di Amazon Redshift

BigQuery Per modificare le impostazioni di ottimizzazione delle conversioni di Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Google BigQuery, quindi scegli Google BigQuery — Amazon Redshift. Nel riquadro a sinistra, scegli Strategie di ottimizzazione. AWS SCT visualizza le impostazioni di ottimizzazione della conversione BigQuery per la conversione in Amazon Redshift.

BigQuery alle impostazioni di ottimizzazione delle conversioni di Amazon Redshift in AWS SCT includono opzioni per quanto segue:

- Per lavorare con l'ottimizzazione automatica delle tabelle. A tale scopo, seleziona Utilizza l'ottimizzazione automatica della tabella di Amazon Redshift.

L'ottimizzazione automatica delle tabelle è un processo di ottimizzazione automatica in Amazon Redshift che ottimizza automaticamente la progettazione delle tabelle. Per ulteriori informazioni, consulta [Working with Automatic Tabelle Optimization](#) nella Amazon Redshift Database Developer Guide.

Per fare affidamento solo sull'ottimizzazione automatica della tabella, scegli Nessuno per la strategia di selezione iniziale delle chiavi.

- Per scegliere le chiavi di ordinamento e distribuzione utilizzando la tua strategia.

Puoi scegliere le chiavi di ordinamento e distribuzione utilizzando i metadati di Amazon Redshift, le informazioni statistiche o entrambe queste opzioni. Per la strategia di selezione iniziale delle chiavi nella scheda Strategie di ottimizzazione, scegli una delle seguenti opzioni:

- Usa i metadati, ignora le informazioni statistiche
- Ignora i metadati, utilizza le informazioni statistiche
- Usa metadati e informazioni statistiche

A seconda dell'opzione scelta, puoi selezionare strategie di ottimizzazione. Quindi, per ogni strategia, inserisci il valore (0-100). Questi valori definiscono il peso di ogni strategia. Utilizzando questi valori di peso, AWS SCT definisce in che modo ciascuna regola influisce sulla scelta delle chiavi di distribuzione e ordinamento. I valori predefiniti si basano sulle best practice di AWS migrazione.

È possibile definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle di piccole dimensioni. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserite il numero minimo e massimo di righe in una tabella per definirla come una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

- Per configurare i dettagli della strategia.

Oltre a definire il peso di ogni strategia di ottimizzazione, puoi configurare le impostazioni di ottimizzazione. Per farlo, scegli Ottimizzazione delle conversioni.

- In Limite colonne chiave di ordinamento, immettete il numero massimo di colonne nella chiave di ordinamento.
- In Valore di soglia inclinato, inserisci la percentuale (0-100) di un valore inclinato per una colonna. AWS SCT esclude le colonne con un valore di inclinazione superiore alla soglia dall'elenco dei candidati per la chiave di distribuzione. AWS SCT definisce il valore distorto per una colonna come il rapporto percentuale tra il numero di occorrenze del valore più comune e il numero totale di record.
- Per le prime N query della tabella della cronologia delle query, inserisci il numero (1—100) delle query utilizzate più di frequente da analizzare.
- In Seleziona utente delle statistiche, scegli l'utente del database per il quale desideri analizzare le statistiche della query.

Inoltre, nella scheda Strategie di ottimizzazione, puoi definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle piccole. Per Numero minimo di righe della tabella e

Numero massimo di righe della tabella, inserisci il numero minimo e massimo di righe in una tabella per considerarla una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

Utilizzo del database Greenplum come fonte per AWS SCT

Puoi utilizzarlo AWS SCT per convertire schemi, oggetti di codice e codice applicativo da Greenplum Database ad Amazon Redshift.

Privilegi per il database Greenplum come fonte

Per utilizzare il database Greenplum come fonte sono necessari i seguenti privilegi:

- CONNECT ON DATABASE *<database_name>*
- USAGE ON SCHEMA *<schema_name>*
- SELEZIONA ATTIVA *<schema_name>*. *<table_name>*
- SELEZIONA UNA SEQUENZA *<schema_name>*. *<sequence_name>*

Nell'esempio precedente, sostituisci i segnaposti come segue:

- Sostituisci *database_name* con il nome del database di origine.
- Sostituisci *schema_name* con il nome dello schema di origine.
- Sostituisci *table_name* con il nome della tabella di origine.
- Sostituisci *sequence_name* con il nome del nome della sequenza.

Connessione al database Greenplum come fonte

Usa la seguente procedura per connetterti al tuo database di origine Greenplum con AWS SCT

Per connetterti al database di origine Greenplum

1. Nel AWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli SAP ASE, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.

4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:

- Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:

1. Per AWSSecret, scegli il nome del segreto.
2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le credenziali del database di origine Greenplum, utilizza le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Inserisci il nome DNS (Domain Name System) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
Database	Immetti il nome del database Greenplum.
User name (Nome utente) e Password (Password)	<p>Inserisci le credenziali del database per connetterti al server del database di origine.</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p>

Parametro	Operazione
Use SSL (Usa SSL)	<p>Scegli questa opzione per utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none"> • Verifica il certificato del server: seleziona questa opzione per verificare il certificato del server utilizzando un archivio attendibile. • Trust store: la posizione di un trust store contenente certificati.
Memorizza la password	<p>AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza dover inserire la password.</p>
Percorso del driver del database Greenplum	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

5. Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
6. Scegli Connetti per connetterti al tuo database di origine.

Impostazioni di conversione da Greenplum ad Amazon Redshift

Per modificare le impostazioni di conversione da Greenplum ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Greenplum, quindi scegli Greenplum — Amazon Redshift. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da Greenplum ad Amazon Redshift.

Le impostazioni di conversione da Greenplum ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungi commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per impostare il numero massimo di tabelle che AWS SCT possono essere applicate al cluster Amazon Redshift di destinazione.

Per il numero massimo di tabelle per il cluster Amazon Redshift di destinazione, scegli il numero di tabelle che AWS SCT possono essere applicate al tuo cluster Amazon Redshift.

Amazon Redshift prevede quote che limitano le tabelle di utilizzo per diversi tipi di nodi del cluster. Se scegli Auto, AWS SCT determina il numero di tabelle da applicare al cluster Amazon Redshift di destinazione in base al tipo di nodo. Facoltativamente, scegli il valore manualmente. Per ulteriori informazioni, consulta [Quote e limiti in Amazon Redshift](#) nella Guida alla gestione di Amazon Redshift.

AWS SCT converte tutte le tabelle di origine, anche se sono più di quelle che il tuo cluster Amazon Redshift può archiviare. AWS SCT memorizza il codice convertito nel progetto e non lo applica al database di destinazione. Se raggiungi la quota del cluster Amazon Redshift per le tabelle quando applichi il codice convertito, viene AWS SCT visualizzato un messaggio di avviso. Inoltre, AWS SCT applica le tabelle al cluster Amazon Redshift di destinazione fino a quando il numero di tabelle raggiunge il limite.

- Per migrare le partizioni della tabella di origine in tabelle separate in Amazon Redshift. A tale scopo, seleziona Usa la vista UNION ALL e inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per una singola tabella di origine.

Amazon Redshift non supporta il partizionamento delle tabelle. Per emulare questo comportamento e velocizzare l'esecuzione delle query, AWS SCT puoi migrare ogni partizione della tabella di origine in una tabella separata in Amazon Redshift. Quindi, AWS SCT crea una vista che include i dati di tutte queste tabelle.

AWS SCT determina automaticamente il numero di partizioni nella tabella di origine. A seconda del tipo di partizionamento della tabella di origine, questo numero può superare la quota delle tabelle che puoi applicare al tuo cluster Amazon Redshift. Per evitare di raggiungere questa quota, inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per le partizioni di una singola tabella di origine. L'opzione predefinita è 368 tabelle, che rappresenta una partizione per 366 giorni all'anno e due tabelle per NO RANGE e partizioni. UNKNOWN

- Per applicare la compressione alle colonne della tabella Amazon Redshift. A tale scopo, seleziona Usa la codifica di compressione.

AWS SCT assegna automaticamente la codifica di compressione alle colonne utilizzando l'algoritmo Amazon Redshift predefinito. Per ulteriori informazioni, consulta [Codifiche di compressione](#) nella Amazon Redshift Database Developer Guide.

Per impostazione predefinita, Amazon Redshift non applica la compressione alle colonne definite come chiavi di ordinamento e distribuzione. È possibile modificare questo comportamento e applicare la compressione a queste colonne. A tale scopo, seleziona Usa la codifica di compressione per le colonne KEY. È possibile selezionare questa opzione solo quando si seleziona l'opzione Usa codifica di compressione.

Impostazioni di ottimizzazione della conversione da Greenplum ad Amazon Redshift

Per modificare le impostazioni di ottimizzazione della conversione da Greenplum ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Greenplum, quindi scegli Greenplum — Amazon Redshift. Nel riquadro a sinistra, scegli Strategie di ottimizzazione. AWS SCT visualizza le impostazioni di ottimizzazione della conversione per la conversione da Greenplum ad Amazon Redshift.

Le impostazioni di ottimizzazione della conversione da Greenplum ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per lavorare con l'ottimizzazione automatica delle tabelle. A tale scopo, seleziona Utilizza l'ottimizzazione automatica della tabella di Amazon Redshift.

L'ottimizzazione automatica delle tabelle è un processo di ottimizzazione automatica in Amazon Redshift che ottimizza automaticamente la progettazione delle tabelle. Per ulteriori informazioni, consulta [Working with Automatic Table Optimization](#) nella Amazon Redshift Database Developer Guide.

Per fare affidamento solo sull'ottimizzazione automatica della tabella, scegli Nessuno per la strategia di selezione iniziale delle chiavi.

- Per scegliere le chiavi di ordinamento e distribuzione utilizzando la tua strategia.

Puoi scegliere le chiavi di ordinamento e distribuzione utilizzando i metadati di Amazon Redshift, le informazioni statistiche o entrambe queste opzioni. Per la strategia di selezione iniziale delle chiavi nella scheda Strategie di ottimizzazione, scegli una delle seguenti opzioni:

- Usa i metadati, ignora le informazioni statistiche
- Ignora i metadati, utilizza le informazioni statistiche
- Usa metadati e informazioni statistiche

A seconda dell'opzione scelta, puoi selezionare strategie di ottimizzazione. Quindi, per ogni strategia, inserisci il valore (0-100). Questi valori definiscono il peso di ogni strategia. Utilizzando questi valori di peso, AWS SCT definisce in che modo ciascuna regola influisce sulla scelta delle chiavi di distribuzione e ordinamento. I valori predefiniti si basano sulle best practice di AWS migrazione.

È possibile definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle di piccole dimensioni. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserite il numero minimo e massimo di righe in una tabella per definirla come una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

- Per configurare i dettagli della strategia.

Oltre a definire il peso di ogni strategia di ottimizzazione, puoi configurare le impostazioni di ottimizzazione. Per farlo, scegli Ottimizzazione delle conversioni.

- In Limite colonne chiave di ordinamento, immettete il numero massimo di colonne nella chiave di ordinamento.
- In Valore di soglia inclinato, inserisci la percentuale (0-100) di un valore inclinato per una colonna. AWS SCT esclude le colonne con un valore di inclinazione superiore alla soglia dall'elenco dei candidati per la chiave di distribuzione. AWS SCT definisce il valore distorto per una colonna come il rapporto percentuale tra il numero di occorrenze del valore più comune e il numero totale di record.
- Per le prime N query della tabella della cronologia delle query, inserisci il numero (1—100) delle query utilizzate più di frequente da analizzare.

- In **Seleziona utente delle statistiche**, scegli l'utente del database per il quale desideri analizzare le statistiche della query.

Inoltre, nella scheda **Strategie di ottimizzazione**, puoi definire la dimensione delle tabelle di piccole dimensioni per la strategia **Trova tabelle piccole**. Per **Numero minimo di righe della tabella** e **Numero massimo di righe della tabella**, inserisci il numero minimo e massimo di righe in una tabella per considerarla una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

Usare Netezza come fonte per AWS SCT

Puoi utilizzare AWS SCT per convertire schemi, oggetti di codice e codice applicativo da Netezza ad Amazon Redshift.

Privilegi per Netezza come fonte

Per utilizzare Netezza come fonte sono necessari i seguenti privilegi:

- seleziona nella vista `system.definition_schema.system`
- seleziona nella tabella `system.definition_schema.system`
- seleziona nella tabella `system.definition_schema.management`
- elenco su `<database_name>`
- elenco su `<schema_name>`
- elenco su `<database_name>.all.table`
- elenco nella `<database_name>tabella.all.external`
- elenco su `<database_name>.all.view`
- elenco nella vista `<database_name>.all.materialized`
- elenco su `<database_name>.all.procedure`
- elenco su `<database_name>.all.sequence`
- elenco su `<database_name>.all.function`
- elenco su `<database_name>.all.aggregate`

Nell'esempio precedente, sostituisci i segnaposti come segue:

- Sostituisci *database_name* con il nome del database di origine.

- Sostituisci *schema_name* con il nome dello schema di origine.

AWS SCT richiede l'accesso alle seguenti tabelle e viste di sistema. È possibile concedere l'accesso a questi oggetti anziché concedere l'accesso a `system.definition_schema.system view` e `system.definition_schema.system tables` nell'elenco precedente.

- seleziona su `system.definition_schema._t_aggregato`
- seleziona su `system.definition_schema._t_class`
- seleziona su `system.definition_schema._t_vincolo`
- seleziona su `system.definition_schema._t_const_relattr`
- seleziona su `system.definition_schema._t_banca_dati`
- seleziona su `system.definition_schema._t_grpobj_priv`
- seleziona su `system.definition_schema._t_grupps`
- seleziona su `system.definition_schema._t_hist_config`
- seleziona su `system.definition_schema._t_oggetto`
- seleziona su `system.definition_schema._t_object_classes`
- seleziona su `system.definition_schema._t_proc`
- seleziona su `system.definition_schema._t_type`
- seleziona su `system.definition_schema._t_utente`
- seleziona su `system.definition_schema._it_usrobj_priv`
- seleziona su `system.definition_schema.sequenza_vt`
- seleziona su `system.definition_schema._v_aggregato`
- seleziona su `system.definition_schema._v_constraint_dipende`
- seleziona su `system.definition_schema._v_database`
- seleziona su `system.definition_schema.tipo di dati_v`
- seleziona su `system.definition_schema._v_dslice`
- seleziona su `system.definition_schema.funzione_v`
- seleziona su `system.definition_schema._v_gruppo`
- seleziona su `system.definition_schema.Relazione_v_obj_`
- seleziona su `system.definition_schema._v_obj_relation_xdb`

- seleziona system.definition_schema. _v_procedura
- seleziona system.definition_schema. _v_relazione_colonna
- seleziona system.definition_schema. _v_relation_dati chiave
- seleziona system.definition_schema. classi _v_relobjm
- seleziona system.definition_schema. _v_schema_xdb
- seleziona system.definition_schema. _v_sequenza
- seleziona system.definition_schema. _v_sinonimo
- seleziona system.definition_schema. _v_informazioni di sistema
- seleziona system.definition_schema. vincolo _v_sys_
- seleziona system.definition_schema. _v_sys_object_dslice_info
- seleziona system.definition_schema. _v_sys_user
- seleziona system.definition_schema. _v_tabella
- seleziona system.definition_schema. _v_table_constraint
- seleziona system.definition_schema. _v_tabella_dist_mappa
- seleziona system.definition_schema. _v_table_organize_colonna
- seleziona system.definition_schema. _v_tabella_storage_stat
- seleziona system.definition_schema. _v_utente
- seleziona system.definition_schema. _v_view
- seleziona su system.information_schema. _v_relazione_colonna
- seleziona su system.information_schema. _v_tabella
- seleziona su \$hist_column_access_*

Connessione a Netezza come fonte

Utilizza la seguente procedura per connetterti al database di origine Netezza con AWS Schema Conversion Tool.

Per connetterti a un database di origine Netezza

1. NelAWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Netezza, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo **Aggiungi fonte**.

3. In **Nome connessione**, inserisci un nome per il tuo database. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per **AWSSecret**, scegli il nome del segreto.
 2. Scegliete **Compila** per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine Netezza, utilizzare le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Inserisci il nome DNS (Domain Name System) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
User name (Nome utente) e Password (Password)	Inserisci le credenziali del database per connetterti al server del database di origine. AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.
Memorizza la password	AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la

Parametro	Operazione
	password del database e connettersi rapidamente al database senza dover inserire la password.
Percorso del driver Netezza	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

- Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
- Scegli Connetti per connetterti al tuo database di origine.

Configurazione della replica continua dei dati

Dopo aver convertito gli schemi del database Netezza e averli applicati al database Amazon Redshift, puoi migrare i dati con AWS SCT agenti di estrazione dei dati. L'agente estrae i tuoi dati e li carica nel tuo bucket Amazon S3. Puoi quindi utilizzarli AWS SCT per copiare i dati da Amazon S3 ad Amazon Redshift.

Se i dati nel database di origine cambiano durante il processo di migrazione, puoi acquisire le modifiche in corso con AWS SCT i tuoi agenti di estrazione dei dati. Quindi puoi replicare queste modifiche in corso nel tuo database di destinazione dopo aver completato la migrazione iniziale dei dati. Questo processo è chiamato replica continua dei dati o acquisizione dei dati di modifica (CDC).

Per configurare la replica continua dei dati per le migrazioni da Netezza ad Amazon Redshift

- Nel tuo database di origine, crea un database di cronologia. È possibile utilizzare il seguente esempio di codice nell'interfaccia a riga di comando (CLI) di Netezza.

```
nzhistcreatedb -d history_database_name -t query -v 1 -u load_user -o histdb_owner
-p your_password
```

Nell'esempio precedente, sostituisci *history_database_name* con il nome del tuo database di cronologia. Quindi, sostituisci *load_user* con il nome dell'utente che hai definito per caricare i dati della cronologia nel database. Quindi, sostituisci *histdb_owner* con il nome dell'utente che hai definito come proprietario del database di cronologia. Assicurati di aver già creato questo utente e di aver concesso l'CREATE DATABASE autorizzazione. Infine, sostituisci *your_password* con una password sicura.

2. Configura la registrazione della cronologia. A tale scopo, utilizzate il seguente esempio di codice.

```
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 STORAGE LIMIT 25
  LOADRETRY 2 VERSION 1;
```

Nell'esempio precedente, sostituisci *history_configuration_name* e *history_database_name* con i nomi della tua configurazione della cronologia e del tuo database di cronologia. Quindi, sostituisci *load_user* con il nome dell'utente che hai definito per caricare i dati della cronologia nel database. Quindi, sostituisci *your_password* con una password sicura.

3. Concedi le autorizzazioni di lettura per tutte le tabelle nel database della cronologia. È possibile utilizzare il seguente esempio di codice per concedere l'SELECT autorizzazione.

```
GRANT SELECT ON history_database_name.ALL.TABLE TO your_user;
```

Nell'esempio precedente, sostituisci *history_database_name* con il nome del tuo database di cronologia. Quindi, sostituisci *your_user* con il nome dell'utente con autorizzazioni minime per lavorare con il tuo database Netezza. Utilizzi le credenziali di questo utente del database inAWS SCT.

4. Raccogli le statistiche per ogni tabella nello schema di origine per ottenere informazioni sulla cardinalità delle colonne. Puoi usare il seguente comando per generare statistiche nel tuo database di cronologia.

```
GENERATE STATISTICS on "schema_name". "table_name";
```

Nell'esempio precedente, sostituisci *schema_name* e *table_name* con il nome dello schema e della tabella del tuo database.

5. Assicurati di aver completato i prerequisiti eseguendo la seguente query:

```
SELECT COUNT(*)  
FROM history_database_name.history_schema_name."$hist_column_access_N";
```

Nell'esempio precedente, sostituisci *history_database_name* e *history_schema_name* con il nome del database e dello schema della cronologia. Quindi, sostituisci *N* con il numero di versione del tuo database di cronologia. Per ulteriori informazioni sulle versioni del database cronologico, consulta la documentazione [IBM Netezza](#).

6. Installa i tuoi agenti di estrazione dei dati. Per ulteriori informazioni, consulta [Installazione degli agenti di estrazione](#).

Assicurati che il {working.folder} parametro nel settings.properties file per tutte le istanze dell'estrattore punti alla stessa cartella. In questo caso, gli estrattori possono coordinare la sessione CDC e utilizzare un unico punto di transazione per tutte le sottoattività.

7. Registra il tuo agente di estrazione dei dati. Per ulteriori informazioni, consulta [Registrazione degli agenti di estrazione con AWS Schema Conversion Tool](#).
8. Crea la tua attività CDC. Per ulteriori informazioni, consulta [Creazione, esecuzione e monitoraggio di qualsiasi AWS SCT attività](#).
 - a. Apri il tuo progetto in AWS SCT. Nel riquadro a sinistra, scegli la tua tabella di origine. Apri il menu contestuale (clic con il pulsante destro del mouse) e scegli Crea attività locale.
 - b. In Nome attività, inserisci un nome descrittivo per l'attività di migrazione dei dati.
 - c. Per la modalità di migrazione, scegli Estrai, carica e copia.
 - d. Seleziona Abilita CDC.
 - e. Scegli la scheda delle impostazioni CDC e definisci l'ambito e la pianificazione delle sessioni CDC.
 - f. Scegli Test task per verificare di poterti connettere alla tua cartella di lavoro, al bucket Amazon S3 e al data warehouse Amazon Redshift.
 - g. Scegli Crea per creare la tua attività.
 - h. Scegli la scheda Attività, scegli l'attività dall'elenco e scegli Avvia.
9. L'AWS SCT attività mantiene la coerenza transazionale nel database di destinazione. L'agente di estrazione dei dati replica le transazioni dall'origine in ordine di ID della transazione.

Se interrompi una delle sessioni di migrazione o se fallisce, anche l'elaborazione del CDC si interrompe.

Impostazioni di conversione da Netezza ad Amazon Redshift

Per modificare le impostazioni di conversione da Netezza ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Netezza, quindi scegli Netezza — Amazon Redshift. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da Netezza ad Amazon Redshift.

Le impostazioni di conversione da Netezza ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungere commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per impostare il numero massimo di tabelle che AWS SCT possono essere applicate al cluster Amazon Redshift di destinazione.

Per il numero massimo di tabelle per il cluster Amazon Redshift di destinazione, scegli il numero di tabelle che AWS SCT possono essere applicate al tuo cluster Amazon Redshift.

Amazon Redshift prevede quote che limitano le tabelle di utilizzo per diversi tipi di nodi del cluster. Se scegli Auto, AWS SCT determina il numero di tabelle da applicare al cluster Amazon Redshift di destinazione in base al tipo di nodo. Facoltativamente, scegli il valore manualmente. Per ulteriori informazioni, consulta [Quote e limiti in Amazon Redshift](#) nella Guida alla gestione di Amazon Redshift.

AWS SCT converte tutte le tabelle di origine, anche se sono più di quelle che il tuo cluster Amazon Redshift può archiviare. AWS SCT memorizza il codice convertito nel progetto e non lo applica al database di destinazione. Se raggiungi la quota del cluster Amazon Redshift per le tabelle quando applichi il codice convertito, viene visualizzato un messaggio di avviso. Inoltre, AWS

SCT applica le tabelle al cluster Amazon Redshift di destinazione fino a quando il numero di tabelle raggiunge il limite.

- Per applicare la compressione alle colonne della tabella Amazon Redshift. A tale scopo, seleziona Usa la codifica di compressione.

AWS SCT assegna automaticamente la codifica di compressione alle colonne utilizzando l'algoritmo Amazon Redshift predefinito. Per ulteriori informazioni, consulta [Codifiche di compressione](#) nella Amazon Redshift Database Developer Guide.

Per impostazione predefinita, Amazon Redshift non applica la compressione alle colonne definite come chiavi di ordinamento e distribuzione. È possibile modificare questo comportamento e applicare la compressione a queste colonne. A tale scopo, seleziona Usa la codifica di compressione per le colonne KEY. È possibile selezionare questa opzione solo quando si seleziona l'opzione Usa codifica di compressione.

Impostazioni di ottimizzazione della conversione da Netezza ad Amazon Redshift

Per modificare le impostazioni di ottimizzazione della conversione da Netezza ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Netezza, quindi scegli Netezza — Amazon Redshift. Nel riquadro a sinistra, scegli Strategie di ottimizzazione. AWS SCT visualizza le impostazioni di ottimizzazione della conversione per la conversione da Netezza ad Amazon Redshift.

Le impostazioni di ottimizzazione della conversione da Netezza ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per lavorare con l'ottimizzazione automatica delle tabelle. A tale scopo, seleziona Utilizza l'ottimizzazione automatica della tabella di Amazon Redshift.

L'ottimizzazione automatica delle tabelle è un processo di ottimizzazione automatica in Amazon Redshift che ottimizza automaticamente la progettazione delle tabelle. Per ulteriori informazioni, consulta [Working with Automatic Tabelle Optimization](#) nella Amazon Redshift Database Developer Guide.

Per fare affidamento solo sull'ottimizzazione automatica della tabella, scegli Nessuno per la strategia di selezione iniziale delle chiavi.

- Per scegliere le chiavi di ordinamento e distribuzione utilizzando la tua strategia.

Puoi scegliere le chiavi di ordinamento e distribuzione utilizzando i metadati di Amazon Redshift, le informazioni statistiche o entrambe queste opzioni. Per la strategia di selezione iniziale delle chiavi nella scheda Strategie di ottimizzazione, scegli una delle seguenti opzioni:

- Usa i metadati, ignora le informazioni statistiche
- Ignora i metadati, utilizza le informazioni statistiche
- Usa metadati e informazioni statistiche

A seconda dell'opzione scelta, puoi selezionare strategie di ottimizzazione. Quindi, per ogni strategia, inserisci il valore (0-100). Questi valori definiscono il peso di ogni strategia. Utilizzando questi valori di peso, AWS SCT definisce in che modo ciascuna regola influisce sulla scelta delle chiavi di distribuzione e ordinamento. I valori predefiniti si basano sulle best practice di AWS migrazione.

È possibile definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle di piccole dimensioni. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserite il numero minimo e massimo di righe in una tabella per definirla come una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

- Per configurare i dettagli della strategia.

Oltre a definire il peso di ogni strategia di ottimizzazione, puoi configurare le impostazioni di ottimizzazione. Per farlo, scegli Ottimizzazione delle conversioni.

- In Limite colonne chiave di ordinamento, immettete il numero massimo di colonne nella chiave di ordinamento.
- In Valore di soglia inclinato, inserisci la percentuale (0-100) di un valore inclinato per una colonna. AWS SCT esclude le colonne con un valore di inclinazione superiore alla soglia dall'elenco dei candidati per la chiave di distribuzione. AWS SCT definisce il valore distorto per una colonna come il rapporto percentuale tra il numero di occorrenze del valore più comune e il numero totale di record.
- Per le prime N query della tabella della cronologia delle query, inserisci il numero (1—100) delle query utilizzate più di frequente da analizzare.
- In Seleziona utente delle statistiche, scegli l'utente del database per il quale desideri analizzare le statistiche della query.

Inoltre, nella scheda Strategie di ottimizzazione, puoi definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle piccole. Per Numero minimo di righe della tabella e

Numero massimo di righe della tabella, inserisci il numero minimo e massimo di righe in una tabella per considerarla una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

Utilizzo di Oracle Data Warehouse come fonte per AWS SCT

Puoi utilizzarli AWS SCT per convertire schemi, oggetti di codice e codice applicativo da Oracle Data Warehouse ad Amazon Redshift o Amazon Redshift e AWS Glue utilizzarli in combinazione.

Privilegi per Oracle Data Warehouse come fonte

Per utilizzare Oracle Data Warehouse come origine sono necessari i seguenti privilegi:

- connect
- select_catalog_role
- select any dictionary

Connessione a Oracle Data Warehouse come origine

Utilizza la seguente procedura per connetterti al database di origine del data warehouse Oracle con AWS Schema Conversion Tool.

Per connettersi a un database di origine di Oracle Data Warehouse

1. Nel AWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Oracle, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per AWSSecret, scegli il nome del segreto.
 2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al data warehouse di origine Oracle, utilizzare le seguenti istruzioni:

Parametro	Operazione
Type (Tipo)	<p>Scegli il tipo di connessione al database. A seconda del tipo, fornisci le seguenti informazioni aggiuntive:</p> <ul style="list-style-type: none">• SID<ul style="list-style-type: none">• Nome server: il nome DNS (Domain Name System) o l'indirizzo IP del server del database di origine.• Server port (Porta del server): la porta utilizzata per la connessione al server del database di origine.• Oracle SID (SID Oracle): l'ID del sistema Oracle (SID). Per individuare il SID Oracle, invia la seguenti query al database Oracle: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre>• Nome del servizio<ul style="list-style-type: none">• Server name (Nome del server): il nome DNS o l'indirizzo IP del server del database di origine.• Server port (Porta del server): la porta utilizzata per la connessione al server del database di origine.• Service Name: (Nome del servizio): il nome del servizio Oracle a cui connetterti.• Alias TNS<ul style="list-style-type: none">• TNS file path (Percorso del file TNS): il percorso al file che contiene le informazioni sulla connessione al nome del Transparent Network Substrate (TNS).• TNS file path (Percorso al file TNS): l'alias TNS del file da utilizzare per la connessione al database di origine.• Identificatore di connessione TNS<ul style="list-style-type: none">• Identificatore di connessione TNS: l'identificatore per le informazioni di connessione TNS registrate.

Parametro	Operazione
User name (Nome utente) e Password (Password)	<p>Inserisci le credenziali del database per connetterti al server del database di origine.</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p>
Use SSL (Usa SSL)	<p>Scegli questa opzione per utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none">• Autenticazione SSL: seleziona questa opzione per utilizzare l'autenticazione SSL per la connessione.• Trust store: la posizione di un trust store contenente certificati.• Archivio chiavi: la posizione di un archivio di chiavi contenente una chiave privata e certificati. Questo valore è obbligatorio se è selezionata l'autenticazione SSL ed è altrimenti facoltativo.
Memorizza la password	<p>AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza dover inserire la password.</p>

Parametro	Operazione
Percorso del driver Oracle	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

- Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
- Scegli Connetti per connetterti al tuo database di origine.

Impostazioni di conversione da Oracle Data Warehouse ad Amazon Redshift

Per modificare le impostazioni di conversione da Oracle Data Warehouse ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Oracle, quindi scegli Oracle — Amazon Redshift. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da Oracle Data Warehouse ad Amazon Redshift.

Le impostazioni di conversione da Oracle Data Warehouse ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungere commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per impostare il numero massimo di tabelle che AWS SCT possono essere applicate al cluster Amazon Redshift di destinazione.

Per il numero massimo di tabelle per il cluster Amazon Redshift di destinazione, scegli il numero di tabelle che AWS SCT possono essere applicate al tuo cluster Amazon Redshift.

Amazon Redshift prevede quote che limitano le tabelle di utilizzo per diversi tipi di nodi del cluster. Se scegli Auto, AWS SCT determina il numero di tabelle da applicare al cluster Amazon Redshift di destinazione in base al tipo di nodo. Facoltativamente, scegli il valore manualmente. Per ulteriori informazioni, consulta [Quote e limiti in Amazon Redshift](#) nella Guida alla gestione di Amazon Redshift.

AWS SCT converte tutte le tabelle di origine, anche se sono più di quelle che il tuo cluster Amazon Redshift può archiviare. AWS SCT memorizza il codice convertito nel progetto e non lo applica al database di destinazione. Se raggiungi la quota del cluster Amazon Redshift per le tabelle quando applichi il codice convertito, viene AWS SCT visualizzato un messaggio di avviso. Inoltre, AWS SCT applica le tabelle al cluster Amazon Redshift di destinazione fino a quando il numero di tabelle raggiunge il limite.

- Per migrare le partizioni della tabella di origine in tabelle separate in Amazon Redshift. A tale scopo, seleziona Usa la vista UNION ALL e inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per una singola tabella di origine.

Amazon Redshift non supporta il partizionamento delle tabelle. Per emulare questo comportamento e velocizzare l'esecuzione delle query, AWS SCT puoi migrare ogni partizione della tabella di origine in una tabella separata in Amazon Redshift. Quindi, AWS SCT crea una vista che include i dati di tutte queste tabelle.

AWS SCT determina automaticamente il numero di partizioni nella tabella di origine. A seconda del tipo di partizionamento della tabella di origine, questo numero può superare la quota delle tabelle che puoi applicare al tuo cluster Amazon Redshift. Per evitare di raggiungere questa quota, inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per le partizioni di una singola tabella di origine. L'opzione predefinita è 368 tabelle, che rappresenta una partizione per 366 giorni all'anno e due tabelle per NO RANGE e partizioni. UNKNOWN

- Per convertire le funzioni di formattazione dei tipi di dati come TO_CHAR TO_DATE, e TO_NUMBER con elementi in formato data/ora, non supportati da Amazon Redshift. Per impostazione predefinita, AWS SCT utilizza le funzioni del pacchetto di estensione per emulare l'utilizzo di questi elementi di formato non supportati nel codice convertito.

Il modello di formato data/ora in Oracle include più elementi rispetto alle stringhe di formato datetime in Amazon Redshift. Quando il codice sorgente include solo elementi in formato data/ora supportati da Amazon Redshift, non sono necessarie le funzioni del pacchetto di estensione nel codice convertito. Per evitare di utilizzare le funzioni del pacchetto di estensione nel codice convertito, seleziona gli elementi del formato Datetype che usi nel codice Oracle sono simili alle

stringhe di formato datetime in Amazon Redshift. In questo caso, il codice convertito funziona più velocemente.

Il modello di formato numerico in Oracle include più elementi rispetto alle stringhe di formato numerico in Amazon Redshift. Quando il codice sorgente include solo elementi in formato numerico supportati da Amazon Redshift, non hai bisogno delle funzioni del pacchetto di estensione nel codice convertito. Per evitare di utilizzare le funzioni del pacchetto di estensione nel codice convertito, seleziona Gli elementi in formato numerico che usi nel codice Oracle sono simili alle stringhe di formato numerico in Amazon Redshift. In questo caso, il codice convertito funziona più velocemente.

- Per convertire Oracle LEAD e funzioni LAG analitiche. Per impostazione predefinita, AWS SCT genera un'azione per ciascuna LEAD LAG funzione.

Quando il tuo codice sorgente non utilizza i valori predefiniti per l'offset in queste funzioni, AWS SCT puoi emulare l'uso di queste funzioni con la NVL funzione. A tale scopo, selezionare Usa la funzione NVL per emulare il comportamento delle funzioni Oracle LEAD e LAG.

- Per emulare il comportamento delle chiavi primarie e univoche nel tuo cluster Amazon Redshift, seleziona Emula il comportamento delle chiavi primarie e univoche.

Amazon Redshift non applica chiavi uniche e primarie e le utilizza solo a scopo informativo. Se usi questi vincoli nel tuo codice, assicurati che AWS SCT emuli il loro comportamento nel codice convertito.

- Per applicare la compressione alle colonne della tabella Amazon Redshift. A tale scopo, seleziona Usa la codifica di compressione.

AWS SCT assegna automaticamente la codifica di compressione alle colonne utilizzando l'algoritmo Amazon Redshift predefinito. Per ulteriori informazioni, consulta [Codifiche di compressione](#) nella Amazon Redshift Database Developer Guide.

Per impostazione predefinita, Amazon Redshift non applica la compressione alle colonne definite come chiavi di ordinamento e distribuzione. È possibile modificare questo comportamento e applicare la compressione a queste colonne. A tale scopo, seleziona Usa la codifica di compressione per le colonne KEY. È possibile selezionare questa opzione solo quando si seleziona l'opzione Usa codifica di compressione.

Impostazioni di ottimizzazione della conversione da Oracle Data Warehouse ad Amazon Redshift

Per modificare le impostazioni di ottimizzazione della conversione da Oracle Data Warehouse ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Oracle, quindi scegli Oracle — Amazon Redshift. Nel riquadro a sinistra, scegli Strategie di ottimizzazione. AWS SCT visualizza le impostazioni di ottimizzazione della conversione per la conversione da Oracle Data Warehouse ad Amazon Redshift.

Le impostazioni di ottimizzazione della conversione da Oracle Data Warehouse ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per lavorare con l'ottimizzazione automatica delle tabelle. A tale scopo, seleziona Utilizza l'ottimizzazione automatica della tabella di Amazon Redshift.

L'ottimizzazione automatica delle tabelle è un processo di ottimizzazione automatica in Amazon Redshift che ottimizza automaticamente la progettazione delle tabelle. Per ulteriori informazioni, consulta [Working with Automatic Table Optimization](#) nella Amazon Redshift Database Developer Guide.

Per fare affidamento solo sull'ottimizzazione automatica della tabella, scegli Nessuno per la strategia di selezione iniziale delle chiavi.

- Per scegliere le chiavi di ordinamento e distribuzione utilizzando la tua strategia.

Puoi scegliere le chiavi di ordinamento e distribuzione utilizzando i metadati di Amazon Redshift, le informazioni statistiche o entrambe queste opzioni. Per la strategia di selezione iniziale delle chiavi nella scheda Strategie di ottimizzazione, scegli una delle seguenti opzioni:

- Usa i metadati, ignora le informazioni statistiche
- Ignora i metadati, utilizza le informazioni statistiche
- Usa metadati e informazioni statistiche

A seconda dell'opzione scelta, puoi selezionare strategie di ottimizzazione. Quindi, per ogni strategia, inserisci il valore (0-100). Questi valori definiscono il peso di ogni strategia. Utilizzando questi valori di peso, AWS SCT definisce in che modo ciascuna regola influisce sulla scelta delle chiavi di distribuzione e ordinamento. I valori predefiniti si basano sulle best practice di AWS migrazione.

È possibile definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle di piccole dimensioni. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserite il numero minimo e massimo di righe in una tabella per definirla come una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

- Per configurare i dettagli della strategia.

Oltre a definire il peso di ogni strategia di ottimizzazione, puoi configurare le impostazioni di ottimizzazione. Per farlo, scegli Ottimizzazione delle conversioni.

- In Limite colonne chiave di ordinamento, immettete il numero massimo di colonne nella chiave di ordinamento.
- In Valore di soglia inclinato, inserisci la percentuale (0-100) di un valore inclinato per una colonna. AWS SCT esclude le colonne con un valore di inclinazione superiore alla soglia dall'elenco dei candidati per la chiave di distribuzione. AWS SCT definisce il valore distorto per una colonna come il rapporto percentuale tra il numero di occorrenze del valore più comune e il numero totale di record.
- Per le prime N query della tabella della cronologia delle query, inserisci il numero (1—100) delle query utilizzate più di frequente da analizzare.
- In Seleziona utente delle statistiche, scegli l'utente del database per il quale desideri analizzare le statistiche della query.

Inoltre, nella scheda Strategie di ottimizzazione, puoi definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle piccole. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserisci il numero minimo e massimo di righe in una tabella per considerarla una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

Usare Snowflake come fonte per AWS SCT

Puoi utilizzarlo AWS SCT per convertire schemi, oggetti di codice e codice applicativo da Snowflake ad Amazon Redshift.

Privilegi per Snowflake come database di origine

È possibile creare un ruolo con privilegi e concedere a questo ruolo il nome di un utente utilizzando il SECURITYADMIN ruolo e il contesto della SECURITYADMIN sessione.

L'esempio seguente crea privilegi minimi e li concede all'`min_privs`utente.

```
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;
```

Nell'esempio precedente, sostituisci i segnaposti come segue:

- Sostituisci *role_name* con il nome di un ruolo con privilegi di sola lettura.
- Sostituisci *db_name* con il nome del database di origine.
- Sostituisci *schema_name* con il nome dello schema di origine.
- Sostituisci *datawarehouse_name* con il nome di un data warehouse richiesto.
- Sostituisci *min_privs* con il nome di un utente con privilegi minimi.

I `DEFAULT_WAREHOUSE` parametri `DEFAULT_ROLE` and sono sensibili ai tasti.

Configurazione dell'accesso sicuro ad Amazon S3

Le policy di sicurezza e gestione degli accessi per un bucket Amazon S3 consentono a Snowflake di accedere, leggere e scrivere dati nel bucket S3. Puoi configurare l'accesso sicuro a un bucket Amazon S3 privato utilizzando il tipo di oggetto `SnowflakeSTORAGE INTEGRATION`. Un oggetto di

integrazione dello storage Snowflake delega la responsabilità di autenticazione a un'entità di gestione dell'identità e degli accessi Snowflake.

Per ulteriori informazioni, consulta [Configurazione di un'integrazione di storage con Snowflake per accedere ad Amazon S3 nella documentazione di Snowflake](#).

Connessione a Snowflake come fonte

Utilizza la procedura seguente per connetterti al tuo database di origine con AWS Schema Conversion Tool.

Per connettersi a un database di origine Snowflake

1. Nel AWS Schema Conversion Tool, scegli **Aggiungi fonte**.
2. Scegli **Snowflake**, quindi scegli **Avanti**.

Viene visualizzata la finestra di dialogo **Aggiungi fonte**.

3. In **Nome connessione**, inserisci un nome per il tuo database. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per **AWSSecret**, scegli il nome del segreto.
 2. Scegliete **Compila** per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al data warehouse di origine Snowflake, utilizza le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Inserisci il nome DNS (Domain Name System) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.

Parametro	Operazione
Database	Inserisci il nome del database Snowflake.
User name (Nome utente) e Password (Password)	<p>Inserisci le credenziali del database per connetterti al server del database di origine.</p> <p>AWS SCT memorizza la tua password in un formato crittografato solo se la richiedi esplicitamente.</p>
Use SSL (Usa SSL)	<p>Scegli questa opzione se desideri utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none">• Percorso chiave privata: la posizione di una chiave privata.• Passphrase: la passphrase per la chiave privata. <p>Per ulteriori informazioni sul supporto SSL per Snowflake, consulta Configurare le opzioni di sicurezza per le connessioni.</p>
Memorizza la password	<p>AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Se si imposta questa opzione, è possibile memorizzare la password del database. In questo modo è possibile connettersi rapidamente al database senza dover inserire la password.</p>
Sentiero automobilistico Snowflake	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

5. Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
6. Scegli Connetti per connetterti al tuo database di origine.

Limitazioni per Snowflake come fonte

Di seguito sono riportate le limitazioni relative all'utilizzo di Snowflake come fonte per: AWS SCT

- Gli identificatori di oggetto devono essere univoci nel contesto del tipo di oggetto e dell'oggetto padre:

Database

Gli identificatori dello schema devono essere univoci all'interno di un database.

Schemi

Gli identificatori di oggetti, ad esempio per tabelle e viste, devono essere univoci all'interno di uno schema.

Tabelle/visualizzazioni

Gli identificatori di colonna devono essere univoci all'interno di una tabella.

- Il numero massimo di tabelle per i tipi di nodi cluster large e xlarge è 9.900. Per i tipi di nodi cluster 8xlarge, il numero massimo di tabelle è 100.000. Il limite include tabelle temporanee, sia definite dall'utente che create da Amazon Redshift durante l'elaborazione delle query o la manutenzione del sistema. Per informazioni, consultare [Quote di Amazon Redshift](#) nella Guida alla gestione dei cluster Amazon Redshift.
- Per le procedure memorizzate, il numero massimo di argomenti di input e output è 32.

Tipi di dati di origine per Snowflake

Di seguito, puoi trovare i tipi di dati di origine Snowflake supportati durante l'utilizzo AWS SCT e la mappatura predefinita su un target Amazon Redshift.

Tipi di dati Snowflake	Tipi di dati Amazon Redshift
NUMBER	NUMERICO (38)
NUMBER(p)	Se p è =< 4, allora SMALLINT

Tipi di dati Snowflake	Tipi di dati Amazon Redshift
	<p>Se p è => 5 e =< 9, allora INTEGER</p> <p>Se p è => 10 e =< 18, allora BIGINT</p> <p>Se p è => 19 allora NUMERIC (p)</p>
NUMERO (p, 0)	<p>Se p è =< 4, allora SMALLINT</p> <p>Se p è => 5 e =< 9, allora INTEGER</p> <p>Se p è => 10 e =< 18, allora BIGINT</p> <p>Se p è => 19 allora: NUMERIC (p,0)</p>
NUMERO (p, s)	Se p è => 1 e =< 38 e se s è => 1 e =< 37, allora NUMERIC(p,s)
FLOAT	FLOAT
TEXT	VARCHAR (MAX)
Caratteri Unicode fino a 16.777.216 byte; fino a 4 byte per carattere.	
TESTO (p)	Se p è =< 65.535 allora, VARCHAR (p)
Caratteri Unicode fino a 65.535 byte; fino a 4 byte per carattere.	
TESTO (p)	Se p è => 65.535 e =< 16.777.216 allora, VARCHAR (MAX)
Caratteri Unicode fino a 16.777.216 byte; fino a 4 byte per carattere.	
BINARY	VARCHAR (MAX)
Caratteri a byte singolo fino a 8.388.608 byte; 1 byte per carattere.	

Tipi di dati Snowflake	Tipi di dati Amazon Redshift
BINARIO (p) Caratteri a byte singolo fino a 65.535 byte; 1 byte per carattere.	VARCHAR (p)
BINARIO (p) Caratteri a byte singolo fino a 8.388.608 byte; 1 byte per carattere.	VARCHAR (MAX)
BOOLEAN	BOOLEAN
DATE	DATE
TIME Valori temporali compresi tra 00:00:00 e 23:59:59.999 999999.	AMMORTIZZARE (18)
TEMPO (f) Valori temporali compresi tra 00:00:00 e 23:59:59.9 (f).	VARCHAR (n) — 9+ dt-attr-1
TIMESTAMP_NTZ	TIMESTAMP
TIMESTAMP_TZ	TIMESTAMPTZ

Impostazioni di conversione da Snowflake ad Amazon Redshift

Per modificare le impostazioni di conversione da Snowflake ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Snowflake, quindi scegli Snowflake — Amazon Redshift. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da Snowflake ad Amazon Redshift.

Le impostazioni di conversione da Snowflake ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungi commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per impostare il numero massimo di tabelle che AWS SCT possono essere applicate al cluster Amazon Redshift di destinazione.

Per il numero massimo di tabelle per il cluster Amazon Redshift di destinazione, scegli il numero di tabelle che AWS SCT possono essere applicate al tuo cluster Amazon Redshift.

Amazon Redshift prevede quote che limitano le tabelle di utilizzo per diversi tipi di nodi del cluster. Se scegli Auto, AWS SCT determina il numero di tabelle da applicare al cluster Amazon Redshift di destinazione in base al tipo di nodo. Facoltativamente, scegli il valore manualmente. Per ulteriori informazioni, consulta [Quote e limiti in Amazon Redshift](#) nella Guida alla gestione di Amazon Redshift.

AWS SCT converte tutte le tabelle di origine, anche se sono più di quelle che il tuo cluster Amazon Redshift può archiviare. AWS SCT memorizza il codice convertito nel progetto e non lo applica al database di destinazione. Se raggiungi la quota del cluster Amazon Redshift per le tabelle quando applichi il codice convertito, viene visualizzato un messaggio di avviso. Inoltre, AWS SCT applica le tabelle al cluster Amazon Redshift di destinazione fino a quando il numero di tabelle raggiunge il limite.

- Per applicare la compressione alle colonne della tabella Amazon Redshift. A tale scopo, seleziona Usa codifica di compressione.

AWS SCT assegna automaticamente la codifica di compressione alle colonne utilizzando l'algoritmo Amazon Redshift predefinito. Per ulteriori informazioni, consulta [Codifiche di compressione](#) nella Amazon Redshift Database Developer Guide.

Per impostazione predefinita, Amazon Redshift non applica la compressione alle colonne definite come chiavi di ordinamento e distribuzione. È possibile modificare questo comportamento e applicare la compressione a queste colonne. A tale scopo, seleziona Usa la codifica di compressione per le colonne KEY. È possibile selezionare questa opzione solo quando si seleziona l'opzione Usa codifica di compressione.

Impostazioni di ottimizzazione della conversione da Snowflake ad Amazon Redshift

Per modificare le impostazioni di ottimizzazione della conversione da Snowflake ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Snowflake, quindi scegli Snowflake — Amazon Redshift. Nel riquadro a sinistra, scegli Strategie di ottimizzazione. AWS SCT visualizza le impostazioni di ottimizzazione della conversione per la conversione da Snowflake ad Amazon Redshift.

Le impostazioni di ottimizzazione della conversione da Snowflake ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per lavorare con l'ottimizzazione automatica delle tabelle. A tale scopo, seleziona Utilizza l'ottimizzazione automatica della tabella di Amazon Redshift.

L'ottimizzazione automatica delle tabelle è un processo di ottimizzazione automatica in Amazon Redshift che ottimizza automaticamente la progettazione delle tabelle. Per ulteriori informazioni, consulta [Working with Automatic Tabelle Optimization](#) nella Amazon Redshift Database Developer Guide.

Per fare affidamento solo sull'ottimizzazione automatica della tabella, scegli Nessuno per la strategia di selezione iniziale delle chiavi.

- Per scegliere le chiavi di ordinamento e distribuzione utilizzando la tua strategia.

Puoi scegliere le chiavi di ordinamento e distribuzione utilizzando i metadati di Amazon Redshift, le informazioni statistiche o entrambe queste opzioni. Per la strategia di selezione iniziale delle chiavi nella scheda Strategie di ottimizzazione, scegli una delle seguenti opzioni:

- Usa i metadati, ignora le informazioni statistiche
- Ignora i metadati, utilizza le informazioni statistiche
- Usa metadati e informazioni statistiche

A seconda dell'opzione scelta, puoi selezionare strategie di ottimizzazione. Quindi, per ogni strategia, inserisci il valore (0-100). Questi valori definiscono il peso di ogni strategia. Utilizzando questi valori di peso, AWS SCT definisce in che modo ciascuna regola influisce sulla scelta delle chiavi di distribuzione e ordinamento. I valori predefiniti si basano sulle best practice di AWS migrazione.

È possibile definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle di piccole dimensioni. Per Numero minimo di righe della tabella e Numero massimo di righe della

tabella, inserite il numero minimo e massimo di righe in una tabella per definirla come una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

- Per configurare i dettagli della strategia.

Oltre a definire il peso di ogni strategia di ottimizzazione, puoi configurare le impostazioni di ottimizzazione. Per farlo, scegli Ottimizzazione delle conversioni.

- In Limite colonne chiave di ordinamento, immettete il numero massimo di colonne nella chiave di ordinamento.
- In Valore di soglia inclinato, inserisci la percentuale (0-100) di un valore inclinato per una colonna. AWS SCT esclude le colonne con un valore di inclinazione superiore alla soglia dall'elenco dei candidati per la chiave di distribuzione. AWS SCT definisce il valore distorto per una colonna come il rapporto percentuale tra il numero di occorrenze del valore più comune e il numero totale di record.
- Per le prime N query della tabella della cronologia delle query, inserisci il numero (1—100) delle query utilizzate più di frequente da analizzare.
- In Seleziona utente delle statistiche, scegli l'utente del database per il quale desideri analizzare le statistiche della query.

Inoltre, nella scheda Strategie di ottimizzazione, puoi definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle piccole. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserisci il numero minimo e massimo di righe in una tabella per considerarla una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

Utilizzo di Microsoft SQL Server Data Warehouse come fonte per AWS SCT

Puoi utilizzarli AWS SCT per convertire schemi, oggetti di codice e codice applicativo da Microsoft SQL Server DW ad Amazon Redshift o Amazon Redshift e utilizzarli in combinazione. AWS Glue

Privilegi per Microsoft SQL Server Data Warehouse come origine

Per utilizzare il data warehouse di Microsoft SQL Server come origine sono necessari i seguenti privilegi:

- VIEW DEFINITION
- VIEW DATABASE STATE

- SELECT ON SCHEMA :: *<schema_name>*

Nell'esempio precedente, sostituisci il `<source_schema>` segnaposto con il nome del sorgente `source_schema`.

Ripeti l'assegnazione per ogni database di cui desideri convertire lo schema.

Inoltre, concedere i seguenti privilegi ed eseguire l'assegnazione per il database master:

- VIEW SERVER STATE

Limitazioni per SQL Server Data Warehouse come origine

L'utilizzo di Microsoft SQL Server Parallel Data Warehouse (PDW) come origine non è attualmente supportato.

Connessione a SQL Server Data Warehouse come origine

Utilizzare la procedura seguente per connettersi al database di origine di SQL Server Data Warehouse con AWS Schema Conversion Tool

Per connettersi a un database di origine di SQL Server Data Warehouse

1. Nel AWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Microsoft SQL Server, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per AWS Secret, scegli il nome del segreto.
 2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al data warehouse di origine di Microsoft SQL Server, utilizzare le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Immetti il nome Domain Name Service (DNS) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
Instance name (Nome dell'istanza)	Immettere il nome dell'istanza per il data warehouse di SQL Server.
User name (Nome utente) e Password (Password)	<p>Inserisci le credenziali del database per connetterti al server del database di origine.</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p>
Use SSL (Usa SSL)	<p>Scegli questa opzione per utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none"> • Certificato server attendibile: seleziona questa opzione per considerare attendibile il certificato del server. • Trust store: un trust store che hai configurato nelle Impostazioni globali.

Parametro	Operazione
Memorizza la password	AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza dover inserire la password.
Percorso del driver SQL Server	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

5. Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
6. Scegli Connetti per connetterti al tuo database di origine.

Impostazioni di conversione da SQL Server Data Warehouse ad Amazon Redshift

Per modificare le impostazioni di conversione da SQL Server Data Warehouse ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Microsoft SQL Server, quindi scegli Microsoft SQL Server — Amazon Redshift. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da SQL Server Data Warehouse ad Amazon Redshift.

Le impostazioni di conversione da SQL Server Data Warehouse ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungere commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per impostare il numero massimo di tabelle che AWS SCT possono essere applicate al cluster Amazon Redshift di destinazione.

Per il numero massimo di tabelle per il cluster Amazon Redshift di destinazione, scegli il numero di tabelle che AWS SCT possono essere applicate al tuo cluster Amazon Redshift.

Amazon Redshift prevede quote che limitano le tabelle di utilizzo per diversi tipi di nodi del cluster. Se scegli Auto, AWS SCT determina il numero di tabelle da applicare al cluster Amazon Redshift di destinazione in base al tipo di nodo. Facoltativamente, scegli il valore manualmente. Per ulteriori informazioni, consulta [Quote e limiti in Amazon Redshift](#) nella Guida alla gestione di Amazon Redshift.

AWS SCT converte tutte le tabelle di origine, anche se sono più di quelle che il tuo cluster Amazon Redshift può archiviare. AWS SCT memorizza il codice convertito nel progetto e non lo applica al database di destinazione. Se raggiungi la quota del cluster Amazon Redshift per le tabelle quando applichi il codice convertito, viene visualizzato un messaggio di avviso. Inoltre, AWS SCT applica le tabelle al cluster Amazon Redshift di destinazione fino a quando il numero di tabelle raggiunge il limite.

- Per migrare le partizioni della tabella di origine in tabelle separate in Amazon Redshift. A tale scopo, seleziona Usa la vista UNION ALL e inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per una singola tabella di origine.

Amazon Redshift non supporta il partizionamento delle tabelle. Per emulare questo comportamento e velocizzare l'esecuzione delle query, AWS SCT puoi migrare ogni partizione della tabella di origine in una tabella separata in Amazon Redshift. Quindi, AWS SCT crea una vista che include i dati di tutte queste tabelle.

AWS SCT determina automaticamente il numero di partizioni nella tabella di origine. A seconda del tipo di partizionamento della tabella di origine, questo numero può superare la quota delle tabelle che puoi applicare al tuo cluster Amazon Redshift. Per evitare di raggiungere questa quota, inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per le partizioni di una singola tabella di origine. L'opzione predefinita è 368 tabelle, che rappresenta una partizione per 366 giorni all'anno e due tabelle per NO RANGE e partizioni. UNKNOWN

- Per applicare la compressione alle colonne della tabella Amazon Redshift. A tale scopo, seleziona Usa codifica di compressione.

AWS SCT assegna automaticamente la codifica di compressione alle colonne utilizzando l'algoritmo Amazon Redshift predefinito. Per ulteriori informazioni, consulta [Codifiche di compressione](#) nella Amazon Redshift Database Developer Guide.

Per impostazione predefinita, Amazon Redshift non applica la compressione alle colonne definite come chiavi di ordinamento e distribuzione. È possibile modificare questo comportamento e applicare la compressione a queste colonne. A tale scopo, seleziona Usa la codifica di compressione per le colonne KEY. È possibile selezionare questa opzione solo quando si seleziona l'opzione Usa codifica di compressione.

Impostazioni di ottimizzazione della conversione da SQL Server Data Warehouse ad Amazon Redshift

Per modificare le impostazioni di ottimizzazione della conversione da SQL Server Data Warehouse ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Microsoft SQL Server, quindi scegli Microsoft SQL Server — Amazon Redshift. Nel riquadro a sinistra, scegli Strategie di ottimizzazione. AWS SCT visualizza le impostazioni di ottimizzazione della conversione per la conversione da SQL Server Data Warehouse ad Amazon Redshift.

Le impostazioni di ottimizzazione della conversione da SQL Server Data Warehouse ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per lavorare con l'ottimizzazione automatica delle tabelle. A tale scopo, seleziona Utilizza l'ottimizzazione automatica della tabella di Amazon Redshift.

L'ottimizzazione automatica delle tabelle è un processo di ottimizzazione automatica in Amazon Redshift che ottimizza automaticamente la progettazione delle tabelle. Per ulteriori informazioni, consulta [Working with Automatic Tabelle Optimization](#) nella Amazon Redshift Database Developer Guide.

Per fare affidamento solo sull'ottimizzazione automatica della tabella, scegli Nessuno per la strategia di selezione iniziale delle chiavi.

- Per scegliere le chiavi di ordinamento e distribuzione utilizzando la tua strategia.

Puoi scegliere le chiavi di ordinamento e distribuzione utilizzando i metadati di Amazon Redshift, le informazioni statistiche o entrambe queste opzioni. Per la strategia di selezione iniziale delle chiavi nella scheda Strategie di ottimizzazione, scegli una delle seguenti opzioni:

- Usa i metadati, ignora le informazioni statistiche
- Ignora i metadati, utilizza le informazioni statistiche
- Usa metadati e informazioni statistiche

A seconda dell'opzione scelta, puoi selezionare strategie di ottimizzazione. Quindi, per ogni strategia, inserisci il valore (0-100). Questi valori definiscono il peso di ogni strategia. Utilizzando questi valori di peso, AWS SCT definisce in che modo ciascuna regola influisce sulla scelta delle chiavi di distribuzione e ordinamento. I valori predefiniti si basano sulle best practice di AWS migrazione.

È possibile definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle di piccole dimensioni. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserite il numero minimo e massimo di righe in una tabella per definirla come una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

- Per configurare i dettagli della strategia.

Oltre a definire il peso di ogni strategia di ottimizzazione, puoi configurare le impostazioni di ottimizzazione. Per farlo, scegli Ottimizzazione delle conversioni.

- In Limite colonne chiave di ordinamento, immettete il numero massimo di colonne nella chiave di ordinamento.
- In Valore di soglia inclinato, inserisci la percentuale (0-100) di un valore inclinato per una colonna. AWS SCT esclude le colonne con un valore di inclinazione superiore alla soglia dall'elenco dei candidati per la chiave di distribuzione. AWS SCT definisce il valore distorto per una colonna come il rapporto percentuale tra il numero di occorrenze del valore più comune e il numero totale di record.
- Per le prime N query della tabella della cronologia delle query, inserisci il numero (1—100) delle query utilizzate più di frequente da analizzare.
- In Seleziona utente delle statistiche, scegli l'utente del database per il quale desideri analizzare le statistiche della query.

Inoltre, nella scheda Strategie di ottimizzazione, puoi definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle piccole. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserisci il numero minimo e massimo di righe in una tabella per considerarla una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

Utilizzo di Teradata come fonte per AWS SCT

Puoi utilizzarli AWS SCT per convertire schemi, oggetti di codice e codice applicativo da Teradata ad Amazon Redshift o Amazon Redshift e utilizzarli in combinazione. AWS Glue

Privilegi per Teradata come fonte

Per utilizzare Teradata come fonte sono necessari i seguenti privilegi:

- SELECT ON DBC
- SELEZIONA SU SYSUDTLIB
- SELEZIONA SU SYSLIB
- SELEZIONA SU <source_database>
- CREA PROCEDURA SU <source_database>

Nell'esempio precedente, sostituisci il <source_database>segnaposto con il nome del database di origine.

AWS SCTrichiede il privilegio CREATE PROCEDURE per eseguire HELP PROCEDURE su tutte le procedure nel database di origine. AWS SCTnon utilizza questo privilegio per creare nuovi oggetti nel database Teradata di origine.

Connessione a Teradata come fonte

Utilizza la seguente procedura per connetterti al database di origine Teradata con AWS Schema Conversion Tool.

Per connetterti a un database di origine Teradata

1. NelAWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Teradata, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCTvisualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:

1. Per AWS Secret, scegli il nome del segreto.
2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine Teradata, utilizzare le seguenti istruzioni:

Parametro	Operazione
Nome della connessione	Inserisci un nome per il tuo database. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.
Server name (Nome del server)	Inserisci il nome DNS (Domain Name System) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
Database	Immetti il nome del database Teradata.
User name (Nome utente) e Password (Password)	<p>Inserisci le credenziali del database per connetterti al server del database di origine.</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p>
Memorizza la password	AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza dover inserire la password.

Parametro	Operazione
Crittografa i dati	Scegliete questa opzione per crittografare i dati che scambiate con il database. Se si sceglie questa opzione, il numero di porta 443 viene utilizzato per trasferire dati crittografati tra il database Teradata AWS SCT e il database Teradata.
Percorso del driver Teradata	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

- Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
- Scegli Connetti per connetterti al tuo database di origine.

Utilizzo dell'autenticazione LDAP con una fonte Teradata

Per configurare l'autenticazione del Lightweight Directory Access Protocol (LDAP) per gli utenti Teradata che eseguono Microsoft Active Directory in Windows, utilizza la seguente procedura.

Nella procedura seguente, il dominio Active Directory è `test.local.com`. Il server Windows è DC ed è configurato con le impostazioni predefinite. Lo script seguente crea l'account `test_ldap` Active Directory e questo account utilizza la `test_ldap` password.

Per configurare l'autenticazione LDAP per gli utenti Teradata che eseguono Microsoft Active Directory in Windows

- Nella directory `/opt/teradata/tdat/tdgss/site`, modifica il file `TdgssUserConfigFile.xml`. Modifica la sezione LDAP come indicato di seguito.

```
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"
```

2. Applica le modifiche eseguendo la configurazione come indicato di seguito.

```
#cd /opt/teradata/tdgss/bin  
#./run_tdgssconfig
```

3. Testa la configurazione eseguendo il seguente comando.

```
# /opt/teradata/tdat/tdgss/14.10.03.01/bin/tdsbind -u test_ldap -w test_ldap
```

L'output visualizzato dovrebbe essere simile al seguente.

```
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. Riavvia TPA utilizzando il seguente comando.

```
#tpareset -f "use updated TDGSSCONFIG GD0"
```

5. Crea lo stesso utente nel database Teradata e in Active Directory, come illustrato di seguito.

```
CREATE USER test_ldap AS PERM=1000, PASSWORD=test_ldap;
```

```
GRANT LOGON ON ALL TO test WITH NULL PASSWORD;
```

Se modifichi la password utente in Active Directory per il tuo utente LDAP, specifica questa nuova password durante la connessione a Teradata in modalità LDAP. In modalità DEFAULT, ti connetti a Teradata utilizzando il nome utente LDAP e qualsiasi password.

Configurazione della raccolta di statistiche nel data warehouse Teradata di origine

Per convertire il data warehouse Teradata di origine, AWS SCT utilizza le statistiche per ottimizzare il data warehouse Amazon Redshift convertito. Puoi raccogliere statistiche nel file delle statistiche AWS SCT o caricarlo. Per ulteriori informazioni, consulta [Raccolta o caricamento di statistiche](#).

Per assicurarti che sia in AWS SCT grado di raccogliere statistiche dal tuo data warehouse, completa le seguenti attività prerequisite.

Per raccogliere statistiche dal tuo data warehouse Teradata

1. Esegui la seguente query per raccogliere le statistiche per tutte le tabelle del tuo data warehouse.

```
collect summary statistics on table_name;
```

Nell'esempio precedente, sostituisci *table_name* con *il nome* della tua tabella di origine. Ripeti la query per ogni tabella che converti.

2. Esegui la seguente query per determinare la stringa di account per l'utente, che utilizzi per convertire il tuo data warehouse.

```
select * from dbc.accountinfo where username = 'user_name'
```

3. Attiva la registrazione delle query per un utente specifico utilizzando la stringa di account dell'esempio precedente.

```
BEGIN QUERY LOGGING WITH OBJECTS, SQL ON ALL ACCOUNT=('$$BUSINESS$$');
```

In alternativa, attiva la registrazione delle query per tutti gli utenti del database.

```
BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT SQLTEXT=0 ON ALL;
```

Dopo aver completato la raccolta delle statistiche del data warehouse, disattiva la registrazione delle query. A tale scopo, è possibile utilizzare il seguente esempio di codice.

```
end query logging with explain, objects, sql on all account=(' $M$BUSI$$$D$H');
```

Raccolta di statistiche in modalità offline dal data warehouse Teradata di origine

Dopo aver configurato la raccolta delle statistiche nel tuo data warehouse Teradata, puoi raccogliere statistiche nel tuo AWS SCT progetto. In alternativa, puoi utilizzare gli script Basic Teradata Query (BTEQ) per raccogliere statistiche in modalità offline. Quindi, puoi caricare i file con le statistiche raccolte nel tuo AWS SCT progetto. Per ulteriori informazioni, consulta [Raccolta o caricamento di statistiche](#).

Per raccogliere statistiche dal data warehouse Teradata in modalità offline

1. Crea lo `off-line_stats.bteq` script con il seguente contenuto.

```
.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. Crea il `td_run_bteq.bat` file che esegue lo script BTEQ creato nel passaggio precedente. Usa il seguente contenuto per questo file.

```
@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. Crea il runme .bat file che esegue il file batch creato nel passaggio precedente. Usa il seguente contenuto per questo file.

```
.\td_run_bteq.bat ServerName UserName DatabaseName
```

Nel runme .bat file *ServerNameUserName*, sostituisci e *DatabaseName* con i valori applicabili.

Quindi, esegui il runme .bat file. Ripeti questo passaggio per ogni data warehouse che converti in Amazon Redshift.

Dopo aver eseguito questo script, riceverai tre file con le statistiche per ogni database. Puoi caricare questi file nel tuo AWS SCT progetto. Per farlo, scegli il tuo data warehouse dal pannello di sinistra del progetto e apri il menu contestuale (clic con il pulsante destro del mouse). Scegli Carica statistiche.

Impostazioni di conversione da Teradata ad Amazon Redshift

Per modificare le impostazioni di conversione da Teradata ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Teradata, quindi scegli Teradata — Amazon Redshift. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da Teradata ad Amazon Redshift.

Le impostazioni di conversione da Teradata ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungi commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per impostare il numero massimo di tabelle che AWS SCT possono essere applicate al cluster Amazon Redshift di destinazione.

Per il numero massimo di tabelle per il cluster Amazon Redshift di destinazione, scegli il numero di tabelle che AWS SCT possono essere applicate al tuo cluster Amazon Redshift.

Amazon Redshift prevede quote che limitano le tabelle di utilizzo per diversi tipi di nodi del cluster. Se scegli Auto, AWS SCT determina il numero di tabelle da applicare al cluster Amazon Redshift di destinazione in base al tipo di nodo. Facoltativamente, scegli il valore manualmente. Per ulteriori informazioni, consulta [Quote e limiti in Amazon Redshift](#) nella Guida alla gestione di Amazon Redshift.

AWS SCT converte tutte le tabelle di origine, anche se sono più di quelle che il tuo cluster Amazon Redshift può archiviare. AWS SCT memorizza il codice convertito nel progetto e non lo applica al database di destinazione. Se raggiungi la quota del cluster Amazon Redshift per le tabelle quando applichi il codice convertito, viene visualizzato un messaggio di avviso. Inoltre, AWS SCT applica le tabelle al cluster Amazon Redshift di destinazione fino a quando il numero di tabelle raggiunge il limite.

- Per migrare le partizioni della tabella di origine in tabelle separate in Amazon Redshift. A tale scopo, seleziona Usa la vista UNION ALL e inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per una singola tabella di origine.

Amazon Redshift non supporta il partizionamento delle tabelle. Per emulare questo comportamento e velocizzare l'esecuzione delle query, AWS SCT puoi migrare ogni partizione della tabella di origine in una tabella separata in Amazon Redshift. Quindi, AWS SCT crea una vista che include i dati di tutte queste tabelle.

AWS SCT determina automaticamente il numero di partizioni nella tabella di origine. A seconda del tipo di partizionamento della tabella di origine, questo numero può superare la quota delle tabelle che puoi applicare al tuo cluster Amazon Redshift. Per evitare di raggiungere questa quota,

inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per le partizioni di una singola tabella di origine. L'opzione predefinita è 368 tabelle, che rappresenta una partizione per 366 giorni all'anno e due tabelle per NO RANGE e partizioni. UNKNOWN

- Per applicare la compressione alle colonne della tabella Amazon Redshift. A tale scopo, seleziona Usa codifica di compressione.

AWS SCT assegna automaticamente la codifica di compressione alle colonne utilizzando l'algoritmo Amazon Redshift predefinito. Per ulteriori informazioni, consulta [Codifiche di compressione](#) nella Amazon Redshift Database Developer Guide.

Per impostazione predefinita, Amazon Redshift non applica la compressione alle colonne definite come chiavi di ordinamento e distribuzione. È possibile modificare questo comportamento e applicare la compressione a queste colonne. A tale scopo, seleziona Usa la codifica di compressione per le colonne KEY. È possibile selezionare questa opzione solo quando si seleziona l'opzione Usa codifica di compressione.

- Per utilizzare un elenco esplicito di colonne nel codice convertito per SELECT * le istruzioni, seleziona Usa dichiarazione esplicita di colonna.
- Per emulare il comportamento delle chiavi primarie e univoche nel tuo cluster Amazon Redshift, seleziona Emula il comportamento delle chiavi primarie e univoche.

Amazon Redshift non applica chiavi uniche e primarie e le utilizza solo a scopo informativo. Se usi questi vincoli nel tuo codice, assicurati che AWS SCT emuli il loro comportamento nel codice convertito.

- Per garantire l'unicità dei dati nelle tabelle Amazon Redshift di destinazione. A tale scopo, seleziona Emula il comportamento delle tabelle SET.

Teradata crea tabelle utilizzando l'elemento di SET sintassi come opzione predefinita. Non puoi aggiungere righe duplicate in una SET tabella. Se il tuo codice sorgente non utilizza questo vincolo di unicità, disattiva questa opzione. In questo caso, il codice convertito funziona più velocemente.

Se il codice sorgente utilizza l'SET opzione nelle tabelle come vincolo di unicità, attiva questa opzione. In questo caso, AWS SCT riscrive INSERT . . SELECT le istruzioni nel codice convertito per emulare il comportamento del database di origine.

Impostazioni di ottimizzazione della conversione da Teradata ad Amazon Redshift

Per modificare le impostazioni di ottimizzazione della conversione da Teradata ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Teradata, quindi scegli Teradata — Amazon Redshift. Nel riquadro a sinistra, scegli Strategie di ottimizzazione. AWS SCT visualizza le impostazioni di ottimizzazione della conversione per la conversione da Teradata ad Amazon Redshift.

Le impostazioni di ottimizzazione della conversione da Teradata ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per lavorare con l'ottimizzazione automatica delle tabelle. A tale scopo, seleziona Utilizza l'ottimizzazione automatica della tabella di Amazon Redshift.

L'ottimizzazione automatica delle tabelle è un processo di ottimizzazione automatica in Amazon Redshift che ottimizza automaticamente la progettazione delle tabelle. Per ulteriori informazioni, consulta [Working with Automatic Tabelle Optimization](#) nella Amazon Redshift Database Developer Guide.

Per fare affidamento solo sull'ottimizzazione automatica della tabella, scegli Nessuno per la strategia di selezione iniziale delle chiavi.

- Per scegliere le chiavi di ordinamento e distribuzione utilizzando la tua strategia.

Puoi scegliere le chiavi di ordinamento e distribuzione utilizzando i metadati di Amazon Redshift, le informazioni statistiche o entrambe queste opzioni. Per la strategia di selezione iniziale delle chiavi nella scheda Strategie di ottimizzazione, scegli una delle seguenti opzioni:

- Usa i metadati, ignora le informazioni statistiche
- Ignora i metadati, utilizza le informazioni statistiche
- Usa metadati e informazioni statistiche

A seconda dell'opzione scelta, puoi selezionare strategie di ottimizzazione. Quindi, per ogni strategia, inserisci il valore (0-100). Questi valori definiscono il peso di ogni strategia. Utilizzando questi valori di peso, AWS SCT definisce in che modo ciascuna regola influisce sulla scelta delle chiavi di distribuzione e ordinamento. I valori predefiniti si basano sulle best practice di AWS migrazione.

È possibile definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle di piccole dimensioni. Per Numero minimo di righe della tabella e Numero massimo di righe della

tabella, inserite il numero minimo e massimo di righe in una tabella per definirla come una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

- Per configurare i dettagli della strategia.

Oltre a definire il peso di ogni strategia di ottimizzazione, puoi configurare le impostazioni di ottimizzazione. Per farlo, scegli Ottimizzazione delle conversioni.

- In Limite colonne chiave di ordinamento, immettete il numero massimo di colonne nella chiave di ordinamento.
- In Valore di soglia inclinato, inserisci la percentuale (0-100) di un valore inclinato per una colonna. AWS SCT esclude le colonne con un valore di inclinazione superiore alla soglia dall'elenco dei candidati per la chiave di distribuzione. AWS SCT definisce il valore distorto per una colonna come il rapporto percentuale tra il numero di occorrenze del valore più comune e il numero totale di record.
- Per le prime N query della tabella della cronologia delle query, inserisci il numero (1—100) delle query utilizzate più di frequente da analizzare.
- In Seleziona utente delle statistiche, scegli l'utente del database per il quale desideri analizzare le statistiche della query.

Inoltre, nella scheda Strategie di ottimizzazione, puoi definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle piccole. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserisci il numero minimo e massimo di righe in una tabella per considerarla una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

Usare Vertica come fonte per AWS SCT

Puoi utilizzarlo AWS SCT per convertire schemi, oggetti di codice e codice applicativo da Vertica ad Amazon Redshift.

Privilegi per Vertica come fonte

Per utilizzare Vertica come fonte sono necessari i seguenti privilegi:

- USAGE ON SCHEMA *<schema_name>*
- USAGE ON SCHEMA PUBLIC
- SELECT ON ALL TABLES IN SCHEMA *<schema_name>*

- SELECT ON ALL SEQUENCES IN SCHEMA *<schema_name>*
- EXECUTE ON ALL FUNCTIONS IN SCHEMA *<schema_name>*
- EXECUTE ON PROCEDURE *<schema_name.procedure_name(procedure_signature)>*

Nell'esempio precedente, sostituisci i segnaposti come segue:

- Sostituisci *schema_name* con il nome dello schema di origine.
- Sostituisci *procedure_name con il nome* di una procedura di origine. Ripeti la concessione per ogni procedura che stai convertendo.
- Sostituisci *procedure_signature* con l'elenco delimitato da virgole dei tipi di argomenti della procedura.

Connessione a Vertica come fonte

Utilizza la seguente procedura per connetterti al database di origine Vertica con AWS Schema Conversion Tool.

Per connetterti a un database di origine Vertica

1. NelAWS Schema Conversion Tool, scegli Aggiungi fonte.
2. Scegli Vertica, quindi scegli Avanti.

Viene visualizzata la finestra di dialogo Aggiungi fonte.

3. In Nome connessione, inserisci un nome per il tuo database. AWS SCTvisualizza questo nome nell'albero nel pannello di sinistra.
4. Usa le credenziali del database AWS Secrets Manager o inseriscile manualmente:
 - Per utilizzare le credenziali del database di Secrets Manager, segui le seguenti istruzioni:
 1. Per AWSSecret, scegli il nome del segreto.
 2. Scegliete Compila per inserire automaticamente tutti i valori nella finestra di dialogo di connessione al database di Secrets Manager.

Per informazioni sull'utilizzo delle credenziali del database di Secrets Manager, vedere [Utilizzo di AWS Secrets Manager](#).

- Per inserire manualmente le informazioni di connessione al database di origine Vertica, utilizza le seguenti istruzioni:

Parametro	Operazione
Server name (Nome del server)	Inserisci il nome DNS (Domain Name System) o l'indirizzo IP del server del database di origine.
Porta del server	Immetti la porta utilizzata per la connessione al server del database di origine.
Database	Immetti il nome del database Vertica.
User name (Nome utente) e Password (Password)	<p>Inserisci le credenziali del database per connetterti al server del database di origine.</p> <p>AWS SCT utilizza la password per connettersi al database di origine solo quando si sceglie di connettersi al database in un progetto. Per evitare di esporre la password per il database di origine, AWS SCT non memorizza la password per impostazione predefinita. Se chiudi il tuo progetto AWS SCT e lo riapri, ti viene richiesta la password per la connessione al database di origine in base alle necessità.</p>
Use SSL (Usa SSL)	<p>Scegli questa opzione per utilizzare Secure Sockets Layer (SSL) per connetterti al tuo database. Fornisci le seguenti informazioni aggiuntive, a seconda dei casi, nella scheda SSL:</p> <ul style="list-style-type: none">• Verifica il certificato del server: scegli questa opzione per verificare il certificato del server utilizzando un archivio attendibile.• Trust store: un trust store che hai configurato nelle Impostazioni globali.• Archivio chiavi: un archivio chiavi che hai configurato nelle Impostazioni globali.

Parametro	Operazione
Memorizza la password	AWS SCT crea un insieme di credenziali sicure per archiviare i certificati SSL e le password del database. Attivando questa opzione, è possibile memorizzare la password del database e connettersi rapidamente al database senza dover inserire la password.
Percorso verticale del driver	<p>Immetti il percorso al driver da utilizzare per la connessione al database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti.</p> <p>Se archivi il percorso del driver nelle impostazioni globali del progetto, il percorso del driver non viene visualizzato nella finestra di dialogo della connessione. Per ulteriori informazioni, consulta Memorizzazione dei percorsi dei driver nelle impostazioni globali.</p>

5. Scegli Test Connection per verificare che AWS SCT possa connettersi al tuo database di origine.
6. Scegli Connetti per connetterti al tuo database di origine.

Impostazioni di conversione da Vertica ad Amazon Redshift

Per modificare le impostazioni di conversione da Vertica ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Vertica, quindi scegli Vertica — Amazon Redshift. AWS SCT visualizza tutte le impostazioni disponibili per la conversione da Vertica ad Amazon Redshift.

Le impostazioni di conversione da Vertica ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per limitare il numero di commenti con le azioni nel codice convertito.

Per Aggiungere commenti nel codice convertito per le azioni con gravità selezionata e superiore, scegli la gravità delle azioni. AWS SCT aggiunge commenti nel codice convertito per le azioni con la gravità selezionata e superiore.

Ad esempio, per ridurre al minimo il numero di commenti nel codice convertito, scegli Solo errori. Per includere commenti per tutte le azioni nel codice convertito, scegli Tutti i messaggi.

- Per impostare il numero massimo di tabelle che AWS SCT possono essere applicate al cluster Amazon Redshift di destinazione.

Per il numero massimo di tabelle per il cluster Amazon Redshift di destinazione, scegli il numero di tabelle che AWS SCT possono essere applicate al tuo cluster Amazon Redshift.

Amazon Redshift prevede quote che limitano le tabelle di utilizzo per diversi tipi di nodi del cluster. Se scegli Auto, AWS SCT determina il numero di tabelle da applicare al cluster Amazon Redshift di destinazione in base al tipo di nodo. Facoltativamente, scegli il valore manualmente. Per ulteriori informazioni, consulta [Quote e limiti in Amazon Redshift](#) nella Guida alla gestione di Amazon Redshift.

AWS SCT converte tutte le tabelle di origine, anche se sono più di quelle che il tuo cluster Amazon Redshift può archiviare. AWS SCT memorizza il codice convertito nel progetto e non lo applica al database di destinazione. Se raggiungi la quota del cluster Amazon Redshift per le tabelle quando applichi il codice convertito, viene visualizzato un messaggio di avviso. Inoltre, AWS SCT applica le tabelle al cluster Amazon Redshift di destinazione fino a quando il numero di tabelle raggiunge il limite.

- Per migrare le partizioni della tabella di origine in tabelle separate in Amazon Redshift. A tale scopo, seleziona Usa la vista UNION ALL e inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per una singola tabella di origine.

Amazon Redshift non supporta il partizionamento delle tabelle. Per emulare questo comportamento e velocizzare l'esecuzione delle query, AWS SCT puoi migrare ogni partizione della tabella di origine in una tabella separata in Amazon Redshift. Quindi, AWS SCT crea una vista che include i dati di tutte queste tabelle.

AWS SCT determina automaticamente il numero di partizioni nella tabella di origine. A seconda del tipo di partizionamento della tabella di origine, questo numero può superare la quota delle tabelle che puoi applicare al tuo cluster Amazon Redshift. Per evitare di raggiungere questa quota, inserisci il numero massimo di tabelle di destinazione che AWS SCT è possibile creare per le partizioni di una singola tabella di origine. L'opzione predefinita è 368 tabelle, che rappresenta una partizione per 366 giorni all'anno e due tabelle per NO RANGE e partizioni. UNKNOWN

- Per applicare la compressione alle colonne della tabella Amazon Redshift. A tale scopo, seleziona Usa codifica di compressione.

AWS SCT assegna automaticamente la codifica di compressione alle colonne utilizzando l'algoritmo Amazon Redshift predefinito. Per ulteriori informazioni, consulta [Codifiche di compressione](#) nella Amazon Redshift Database Developer Guide.

Per impostazione predefinita, Amazon Redshift non applica la compressione alle colonne definite come chiavi di ordinamento e distribuzione. È possibile modificare questo comportamento e applicare la compressione a queste colonne. A tale scopo, seleziona Usa la codifica di compressione per le colonne KEY. È possibile selezionare questa opzione solo quando si seleziona l'opzione Usa codifica di compressione.

Impostazioni di ottimizzazione della conversione da Vertica ad Amazon Redshift

Per modificare le impostazioni di ottimizzazione della conversione da Vertica ad Amazon Redshift, scegli Impostazioni in AWS SCT, quindi scegli Impostazioni di conversione. Dall'elenco superiore, scegli Vertica, quindi scegli Vertica — Amazon Redshift. Nel riquadro a sinistra, scegli Strategie di ottimizzazione. AWS SCT visualizza le impostazioni di ottimizzazione della conversione per la conversione da Vertica ad Amazon Redshift.

Le impostazioni di ottimizzazione delle conversioni da Vertica ad Amazon Redshift AWS SCT includono opzioni per quanto segue:

- Per lavorare con l'ottimizzazione automatica delle tabelle. A tale scopo, seleziona Utilizza l'ottimizzazione automatica della tabella di Amazon Redshift.

L'ottimizzazione automatica delle tabelle è un processo di ottimizzazione automatica in Amazon Redshift che ottimizza automaticamente la progettazione delle tabelle. Per ulteriori informazioni, consulta [Working with Automatic Tabelle Optimization](#) nella Amazon Redshift Database Developer Guide.

Per fare affidamento solo sull'ottimizzazione automatica della tabella, scegli Nessuno per la strategia di selezione iniziale delle chiavi.

- Per scegliere le chiavi di ordinamento e distribuzione utilizzando la tua strategia.

Puoi scegliere le chiavi di ordinamento e distribuzione utilizzando i metadati di Amazon Redshift, le informazioni statistiche o entrambe queste opzioni. Per la strategia di selezione iniziale delle chiavi nella scheda Strategie di ottimizzazione, scegli una delle seguenti opzioni:

- Usa i metadati, ignora le informazioni statistiche

- Ignora i metadati, utilizza le informazioni statistiche
- Usa metadati e informazioni statistiche

A seconda dell'opzione scelta, puoi selezionare strategie di ottimizzazione. Quindi, per ogni strategia, inserisci il valore (0-100). Questi valori definiscono il peso di ogni strategia. Utilizzando questi valori di peso, AWS SCT definisce in che modo ciascuna regola influisce sulla scelta delle chiavi di distribuzione e ordinamento. I valori predefiniti si basano sulle best practice di AWS migrazione.

È possibile definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle di piccole dimensioni. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserisci il numero minimo e massimo di righe in una tabella per definirla come una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

- Per configurare i dettagli della strategia.

Oltre a definire il peso di ogni strategia di ottimizzazione, puoi configurare le impostazioni di ottimizzazione. Per farlo, scegli Ottimizzazione delle conversioni.

- In Limite colonne chiave di ordinamento, immettete il numero massimo di colonne nella chiave di ordinamento.
- In Valore di soglia inclinato, inserisci la percentuale (0-100) di un valore inclinato per una colonna. AWS SCT esclude le colonne con un valore di inclinazione superiore alla soglia dall'elenco dei candidati per la chiave di distribuzione. AWS SCT definisce il valore distorto per una colonna come il rapporto percentuale tra il numero di occorrenze del valore più comune e il numero totale di record.
- Per le prime N query della tabella della cronologia delle query, inserisci il numero (1—100) delle query utilizzate più di frequente da analizzare.
- In Seleziona utente delle statistiche, scegli l'utente del database per il quale desideri analizzare le statistiche della query.

Inoltre, nella scheda Strategie di ottimizzazione, puoi definire la dimensione delle tabelle di piccole dimensioni per la strategia Trova tabelle piccole. Per Numero minimo di righe della tabella e Numero massimo di righe della tabella, inserisci il numero minimo e massimo di righe in una tabella per considerarla una tabella piccola. AWS SCT applica lo stile ALL di distribuzione alle tabelle di piccole dimensioni. In questo caso, una copia dell'intera tabella viene distribuita su ogni nodo.

Creazione di regole di mappatura in AWS SCT

È possibile aggiungere più database di origine e di destinazione in un unico AWS SCT progetto. In questo modo si semplifica la gestione dei progetti, quando si migrano più database su piattaforme di destinazione diverse.

Dopo aver creato un nuovo progetto e aggiunto i database di origine e di destinazione, crea regole di mappatura. AWS SCT richiede almeno una regola di mappatura per creare un rapporto di valutazione della migrazione e convertire gli schemi del database.

Una regola di mappatura descrive una coppia sorgente-destinazione che include uno schema del database di origine o un database di origine e una piattaforma di database di destinazione. È possibile creare più regole di mappatura in un singolo AWS SCT progetto. Usa le regole di mappatura per convertire ogni schema del database di origine nella giusta piattaforma di database di destinazione.

Per modificare il nome dello schema nel codice convertito, imposta una regola di migrazione. Ad esempio, con le regole di migrazione, puoi rinominare lo schema, aggiungere un prefisso ai nomi degli oggetti, modificare le regole di confronto delle colonne o modificare i tipi di dati. Per applicare queste modifiche al codice convertito, assicurati di creare regole di migrazione prima di convertire lo schema di origine. Per ulteriori informazioni, consulta [Creazione di regole di migrazione](#).

È possibile creare regole di mappatura solo per le coppie di conversione del database supportate. Per l'elenco delle coppie di conversione supportate, vedere [Origini per AWS SCT](#).

Se si apre un progetto salvato nella AWS SCT versione 1.0.655 o precedente, crea AWS SCT automaticamente regole di mappatura per tutti gli schemi del database di origine sulla piattaforma del database di destinazione. Per aggiungere altre piattaforme di database di destinazione, elimina le regole di mappatura esistenti e quindi crea nuove regole di mappatura.

Argomenti

- [Aggiungere una nuova regola di mappatura](#)
- [Gestione delle regole di mappatura](#)
- [Utilizzo di obiettivi virtuali](#)
- [Limitazioni all'utilizzo di più server in un singolo AWS SCT progetto](#)

Aggiungere una nuova regola di mappatura

È possibile creare più regole di mappatura in un singolo progetto. AWS SCT salva le regole di mappatura come parte del progetto. Con il progetto aperto, utilizzate la procedura seguente per aggiungere una nuova regola di mappatura.

Creazione delle regole di mappatura

1. Nel menu **Visualizza**, selezionare **Visualizzazione mappatura**.
2. Nel pannello di sinistra, scegli uno schema o un database da aggiungere alla regola di mappatura.
3. Nel pannello di destra, scegli una piattaforma di database di destinazione per lo schema o il database di origine selezionato.

Puoi scegliere una piattaforma di database virtuale come destinazione. Per ulteriori informazioni, consulta [Utilizzo di obiettivi virtuali](#).

4. Scegli **Crea mappatura**.

AWS SCT aggiunge questa nuova regola di mappatura all'elenco delle mappature del server.

Aggiungi regole di mappatura per tutte le coppie di conversione. Per creare un rapporto di valutazione o convertire gli schemi del database, scegliere **Visualizzazione principale** dal menu **Visualizza**.

AWS SCT evidenzia in grassetto tutti gli oggetti dello schema che fanno parte di una regola di mappatura.

Gestione delle regole di mappatura

Puoi filtrare o eliminare le regole di mappatura esistenti e aggiungere una nuova regola di mappatura nel tuo progetto AWS Schema Conversion Tool (AWS SCT).

Quando si crea una regola di mappatura per l'intero database di origine, AWS SCT crea una regola di mappatura per ogni schema del database di origine. Per i progetti che coinvolgono dozzine di schemi o addirittura database, può essere difficile capire quale destinazione viene utilizzata per un determinato schema. Per trovare rapidamente una regola di mappatura per il tuo schema, utilizza una o più delle seguenti opzioni di filtro in AWS SCT.

Per filtrare le regole di mappatura

1. Nel menu Visualizza, selezionare Visualizzazione mappatura.
2. Per i server di origine, scegli il database di origine.

Il filtro predefinito è Tutti, il che significa che AWS SCT visualizza le regole di mappatura per tutti i database di origine.

3. Per Schema di origine, immettere il nome dello schema di origine. Usa la percentuale (%) come jolly per sostituire un numero qualsiasi di simboli nel nome dello schema.

Il filtro predefinito è % wildcard, il che significa che AWS SCT visualizza le regole di mappatura per tutti i nomi degli schemi del database di origine.

4. In Ha regole di migrazione, scegli Sì per visualizzare le regole di mappatura per le quali sono state create le regole di migrazione dei dati. Scegli No per visualizzare le regole di mappatura che non prevedono regole di migrazione dei dati. Per ulteriori informazioni, consulta [Creazione di regole di migrazione dei dati in AWS SCT](#).

Il filtro predefinito è Tutto, il che significa che AWS SCT vengono visualizzate tutte le regole di mappatura.

5. Per i server Target, scegli il database di destinazione.

Il filtro predefinito è Tutti, il che significa che AWS SCT visualizza le regole di mappatura per tutti i database di destinazione.

Con il progetto aperto, utilizzate la procedura seguente per eliminare una regola di mappatura. Per ulteriori informazioni sull'aggiunta di regole di mappatura, vedere [Aggiungere una nuova regola di mappatura](#).

Per eliminare le regole di mappatura

1. Nel menu Visualizza, selezionare Visualizzazione mappatura.
2. Per le mappature dei server, scegli le regole di mappatura da eliminare.
3. Scegliete Elimina le mappature selezionate.

AWS SCT elimina le regole di mappatura selezionate.

Utilizzo di obiettivi virtuali

Puoi vedere come AWS SCT converte lo schema del tuo database di origine in qualsiasi piattaforma di database di destinazione supportata. Per farlo, non è necessario connettersi a un database di destinazione esistente. Invece, puoi scegliere una piattaforma di database di destinazione virtuale nel pannello di destra quando crei una regola di mappatura. Per ulteriori informazioni, consulta [Aggiungere una nuova regola di mappatura](#). Assicurati di espandere i server, i cluster NoSQL e i nodi ETL nel pannello di destra per visualizzare l'elenco delle piattaforme di database di destinazione virtuali.

AWS SCT supporta le seguenti piattaforme di database di destinazione virtuali:

- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon DynamoDB
- Amazon Redshift
- Amazon Redshift e AWS Glue
- AWS Glue
- AWS Glue Studio
- Babelfish per Aurora PostgreSQL
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Se utilizzi Babelfish for Aurora PostgreSQL come piattaforma di database di destinazione, puoi creare solo un rapporto di valutazione della migrazione del database. Per ulteriori informazioni, consulta [the section called “Rapporti di valutazione della migrazione”](#).

Se utilizzi una piattaforma di database di destinazione virtuale, puoi salvare il codice convertito in un file. Per ulteriori informazioni, consulta [the section called “Salvare lo schema convertito”](#).

Limitazioni all'utilizzo di più server in un singolo AWS SCT progetto

Le seguenti limitazioni si applicano quando si convertono schemi utilizzando più server in un unico AWS SCT progetto:

- Puoi aggiungere lo stesso server a un progetto solo una volta.
- Non è possibile mappare gli schemi del server su uno schema di destinazione specifico, ma solo su un server di destinazione. AWS SCT crea lo schema di destinazione durante la conversione.
- Non è possibile mappare oggetti di origine di livello inferiore sul server di destinazione.
- È possibile mappare uno schema di origine a un solo server di destinazione in un progetto.
- Assicurati di mappare un'origine su un server di destinazione per creare un rapporto di valutazione, convertire schemi o estrarre dati.

Creazione di report di conversione

Quando pianifichi una conversione di database, è utile creare dei report per aiutarti a comprenderne le implicazioni. Puoi creare report utilizzando AWS Schema Conversion Tool o .

Puoi utilizzare AWS SCT per creare un report di valutazione della migrazione del database. Con questo report, avrai un riepilogo delle attività di conversione dello schema e i dettagli sugli elementi che non possono essere convertiti automaticamente nel database di destinazione. Puoi usare questo report per valutare quanta parte del progetto può essere completata tramite AWS SCT e cos'altro hai bisogno per completare la conversione. Per creare un report di valutazione, utilizza Create Report (Crea report) dal menu contestuale (pulsante destro del mouse) del database in AWS SCT.

Argomenti

- [Creazione di report di valutazione della migrazione con AWS SCT](#)

Creazione di report di valutazione della migrazione con AWS SCT

Una parte importante di AWS Schema Conversion Tool è il rapporto di valutazione che genera per stimare la complessità della conversione dello schema. Questo rapporto di valutazione della migrazione del database riassume tutte le attività di conversione dello schema e descrive in dettaglio le azioni relative allo schema che non può essere convertito nel motore di database dell'istanza DB di destinazione. Puoi visualizzare il report nell'applicazione o esportarlo come file CSV (Comma-Separated) o PDF.

Se aggiungi più database di origine e destinazione in un singolo progetto, AWS SCT aggrega i report per tutte le coppie di conversione in un unico rapporto di valutazione della migrazione del database.

È possibile utilizzare piattaforme di database di destinazione virtuali per generare un rapporto di valutazione e comprendere la complessità della migrazione verso una piattaforma di database selezionata. In questo caso, non è necessario connettersi alla piattaforma del database di destinazione. Ad esempio, puoi utilizzare Babelfish per Aurora PostgreSQL come piattaforma di database di destinazione virtuale per creare un rapporto di valutazione della migrazione del database. Per ulteriori informazioni sulle piattaforme di database di destinazione virtuali, vedere [the section called “Obiettivi virtuali”](#).

Il report di valutazione della migrazione include i seguenti elementi:

- Sintesi

- Valutazione licenza
- Supporto cloud, che indica le eventuali caratteristiche nel database di origine che non sono disponibili su quello di destinazione.
- Raccomandazioni, incluse conversione di oggetti server, suggerimenti di backup e modifiche al server collegato

Il report include anche le stime dello sforzo necessario per scrivere il codice equivalente per l'istanza database di destinazione che non è stata convertita automaticamente.

SeAWS SCT utilizzi la migrazione dello schema esistente su un'istanza DB di Amazon RDS, puoi utilizzare il report per analizzare i requisiti per il passaggio aAWS cloud e modificare il tipo di licenza.

Argomenti

- [Creazione di un rapporto di valutazione della migrazione del database](#)
- [Visualizzazione del rapporto di valutazione](#)
- [Salvataggio del rapporto di valutazione](#)
- [Configurazione del rapporto di valutazione](#)
- [Creazione di un rapporto di valutazione multiserver per la migrazione del database](#)

Creazione di un rapporto di valutazione della migrazione del database

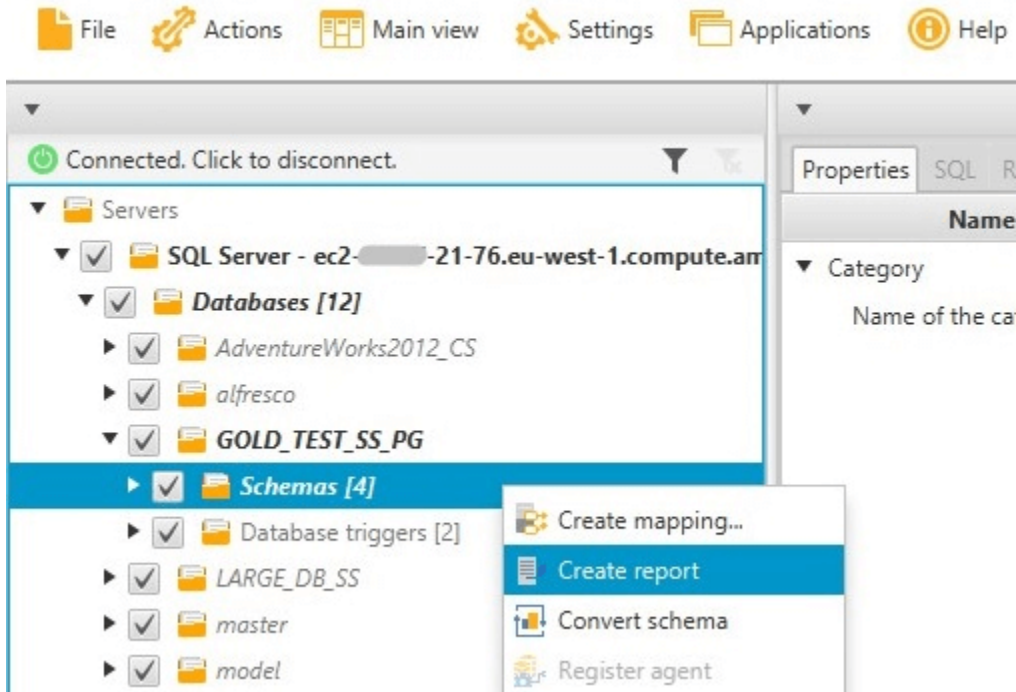
Utilizza la procedura seguente per creare un report di valutazione della migrazione del database.

Creazione di un report di valutazione della migrazione del database

1. Assicurati di aver creato una regola di mappatura per lo schema del database di origine per cui creare un rapporto di valutazione. Per ulteriori informazioni, consulta [Aggiungere una nuova regola di mappatura](#).
2. Nel menu Visualizza, scegli Vista principale.
3. Nel pannello di sinistra che mostra lo schema del database di origine, scegli un oggetto dello schema per cui creare un rapporto di valutazione. Per includere più schemi di database nel rapporto, scegli il nodo principale, ad esempio Schemi.

Assicurati di aver selezionato le caselle di controllo per tutti gli oggetti dello schema per cui creare un rapporto di valutazione.

4. Apri il menu di scelta rapida (tasto destro del mouse) per l'oggetto, quindi scegli Crea report.



Visualizzazione del rapporto di valutazione

Dopo aver creato un report di valutazione, si apre la visualizzazione del report di valutazione, contenente le seguenti schede:

- Riepilogo
- Action Items (Attività)

La scheda Summary (Riepilogo) mostra gli elementi che sono stati convertiti automaticamente oppure non convertiti.

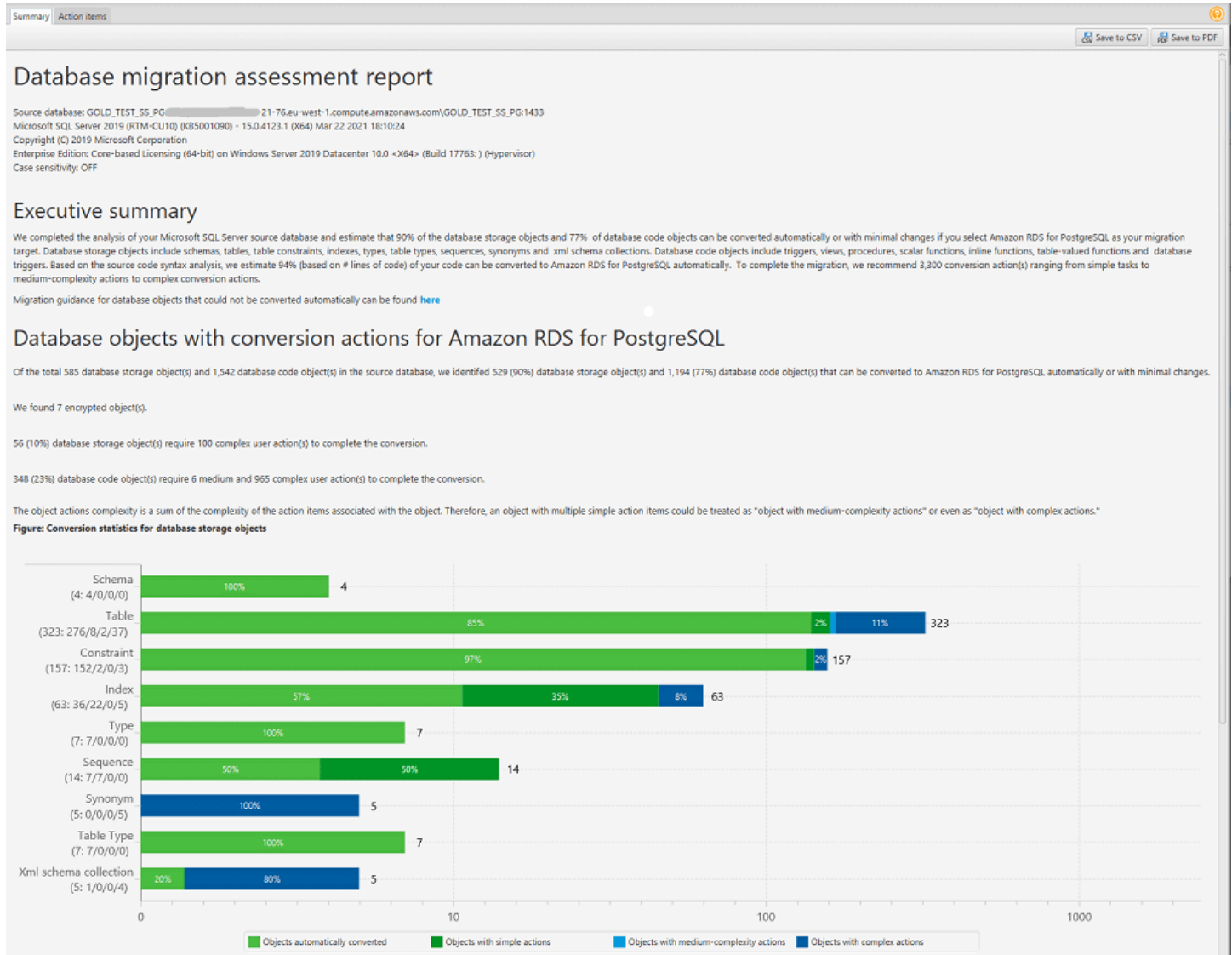
La scheda Action Items (Attività) mostra gli elementi che non sono stati convertiti automaticamente e i suggerimenti sulle operazioni da eseguire.

Argomenti

- [Riepilogo del report di valutazione](#)
- [Elementi d'azione del rapporto di valutazione](#)
- [Messaggio di avviso del rapporto di valutazione](#)

Riepilogo del report di valutazione

Nella scheda Summary (Riepilogo) vengono visualizzate le informazioni di riepilogo del report di valutazione della migrazione del database. Essa mostra gli elementi convertiti automaticamente e quelli non convertiti automaticamente.



Per gli elementi dello schema che non possono essere convertiti automaticamente al motore di database di destinazione, il riepilogo include una stima dello sforzo necessario per creare elementi dello schema nell'istanza database di destinazione che siano equivalenti a quelli presenti nell'istanza database di origine.

Nel report, il tempo stimato per convertire questi elementi dello schema viene suddiviso nelle categorie seguenti:

- **Semplice:** azioni che possono essere completate in meno di due ore.
- **Medio:** azioni più complesse che possono essere completate in 2-6 ore.
- **Significativo:** azioni molto complesse che richiedono più di sei ore per essere completate.

La sezione Valutazione delle licenze e Support cloud contiene informazioni sullo spostamento dello schema di database locale esistente su un'istanza DB Amazon RDS che esegue lo stesso motore. Ad esempio, se intendi modificare i tipi di licenza, questa sezione del report indica quali funzionalità del database corrente devono essere rimosse.

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.

Elementi d'azione del rapporto di valutazione

La visualizzazione del report di valutazione include anche la scheda Action Items (Attività). Questa scheda contiene un elenco di elementi che non possono essere convertiti automaticamente nel motore di database dell'istanza DB Amazon RDS di destinazione. Se selezioni un'attività dall'elenco, AWS SCT evidenzia l'elemento dallo schema al quale si applica l'attività.

Il report contiene anche le raccomandazioni per la conversione manuale dell'elemento dello schema. Ad esempio, dopo l'esecuzione della valutazione, i report dettagliati per il database/lo schema mostrano lo sforzo richiesto per progettare e implementare i consigli per la conversione degli elementi di azione. Per ulteriori informazioni circa la gestione delle conversioni manuali, consulta [Gestione delle conversioni manuali in 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, schemas, and tables. The main pane shows a list of issues, with the following details for Issue 9997:

- 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

Below the issue list, the source SQL code is shown:

```

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

```

The target database is identified as Amazon RDS for MySQL.

Messaggio di avviso del rapporto di valutazione

Per valutare la complessità della conversione in un altro motore di database, è necessario accedere agli oggetti nel database di origine. Quando SCT non è in grado di eseguire una valutazione a causa di problemi durante la scansione, viene emesso un messaggio di avviso che indica che la percentuale di conversione complessiva è ridotta.

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.

Di seguito sono riportati i motivi per cui AWS SCT potrebbero verificarsi problemi durante la scansione:

- L'account utente connesso al database non ha accesso a tutti gli oggetti necessari.
- Un oggetto citato nello schema non esiste più nel database.

- SCT sta cercando di valutare un oggetto crittografato.

Per ulteriori informazioni sulle autorizzazioni e i privilegi di sicurezza richiesti da SCT per il database, consulta [Origini per AWS SCT](#) la sezione appropriata del database di origine in questa guida.

Salvataggio del rapporto di valutazione

Dopo aver [creato un rapporto di valutazione della migrazione del database](#), è possibile salvare una copia locale del rapporto di valutazione della migrazione del database come file PDF o file con valori delimitati da virgole (CSV).

Per salvare un rapporto di valutazione della migrazione del database come file PDF

1. Nel menu in alto, scegli Visualizza, quindi scegli Visualizzazione rapporto di valutazione.
2. Seleziona la scheda Summary (Riepilogo).
3. Scegli Salva in PDF in alto a destra.

Per salvare un rapporto di valutazione della migrazione del database come file CSV

1. Nel menu in alto, scegli Visualizza, quindi scegli Visualizzazione rapporto di valutazione.
2. Seleziona la scheda Summary (Riepilogo).
3. Scegli Salva in CSV in alto a destra.

Il file PDF contiene sia i dati di riepilogo che le informazioni sull'azione, come illustrato nel seguente esempio.

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.

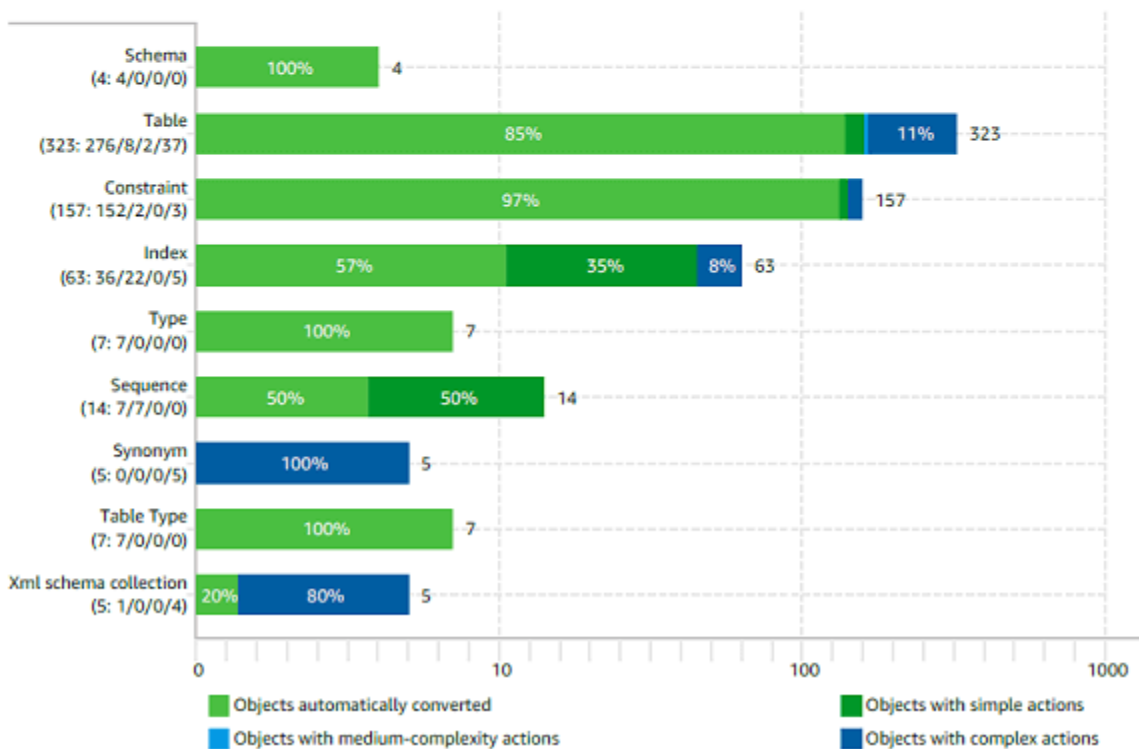
We found 7 encrypted object(s).

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



Quando scegli l'opzione Salva in CSV, AWS SCT crea tre file CSV.

Il primo file CSV contiene le informazioni seguenti relative alle azioni:

- Categoria
- Ricorrenza: il nome del file, il numero di riga e la posizione dell'elemento
- Numero dell'oggetto dell'azione
- Subject
- Group (Gruppo)

- Descrizione
- Riferimenti alla documentazione
- Azione consigliata
- Complessità stimata

Il secondo file CSV include il `Action_Items_Summary` suffisso nel nome e contiene le informazioni sul numero di ricorrenze di tutte le azioni.

Nell'esempio seguente, i valori nella colonna `Sforzo` della curva di apprendimento indicano la quantità di impegno necessaria per progettare un approccio alla conversione di ogni elemento di azione. I valori nella colonna `Impegno` per convertire un'occorrenza dell'azione indicano lo sforzo necessario per convertire ogni azione, seguendo l'approccio progettato. I valori utilizzati per indicare il livello di impegno necessario si basano su una scala ponderata, da basso (minimo) a elevato (massimo).

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

Il terzo file CSV include `Summary` nel suo nome e contiene il seguente riepilogo:

- Categoria
- Numero di oggetti
- Oggetti convertiti automaticamente
- Oggetti con azioni semplici
- Oggetti con azioni di media complessità
- Oggetti con azioni complesse
- Righe totali di codice

Configurazione del rapporto di valutazione

È possibile configurare la quantità di dettagli da AWS SCT includere nei report di valutazione.

Per configurare un rapporto di valutazione della migrazione del database

1. Nel menu Impostazioni, scegli Impostazioni globali, quindi scegli Rapporto di valutazione.
2. Per le ricorrenze delle azioni, scegli Solo i primi cinque problemi per limitare il numero di azioni di un singolo tipo nel rapporto di valutazione. Scegli Tutti i problemi per includere tutte le azioni di ogni tipo nel rapporto di valutazione.
3. Per i file analizzati dagli script SQL, scegli Elenca non più di **X** file per limitare il numero di file di script SQL nel rapporto di valutazione **X**. Inserisci il numero di file. Scegli Elenca tutti i file analizzati per includere tutti i file di script SQL nel rapporto di valutazione.
4. Seleziona Apri report dopo il salvataggio per aprire automaticamente il file dopo aver salvato una copia locale del rapporto di valutazione della migrazione del database. Per ulteriori informazioni, consulta

Dopo aver [creato un rapporto di valutazione della migrazione del database](#), è possibile [salvare una copia locale del rapporto di valutazione della migrazione del database come file PDF o file con valori delimitati da virgole \(CSV\)](#).

Per salvare un rapporto di valutazione della migrazione del database come file PDF

1. Nel menu in alto, scegli Visualizza, quindi scegli Visualizzazione rapporto di valutazione.
 2. Seleziona la scheda Summary (Riepilogo).
 3. Scegli Salva in PDF in alto a destra.
-

Per salvare un rapporto di valutazione della migrazione del database come file CSV

1. Nel menu in alto, scegli Visualizza, quindi scegli Visualizzazione rapporto di valutazione.
 2. Seleziona la scheda Summary (Riepilogo).
 3. Scegli Salva in CSV in alto a destra.
-

Il file PDF contiene sia i dati di riepilogo che le informazioni sull'azione, come illustrato nel seguente esempio.

Database objects with conversion actions for Amazon RDS for PostgreSQL

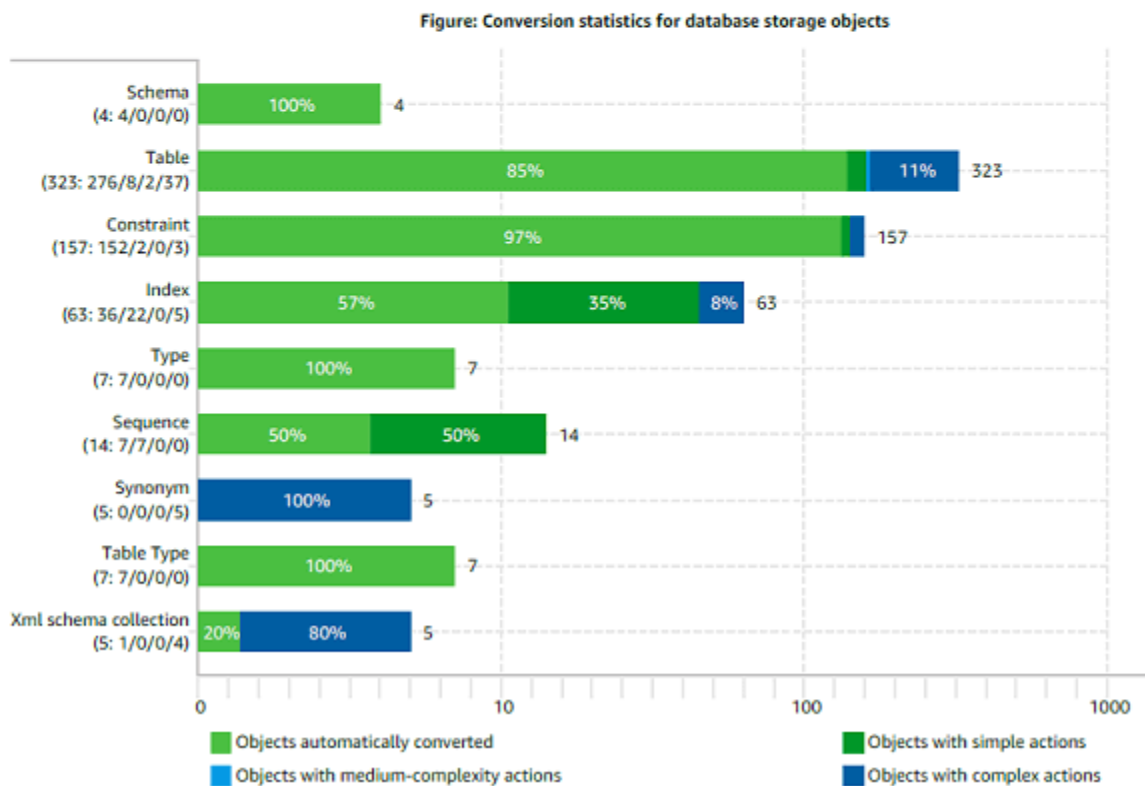
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

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."



Quando scegli l'opzione Salva in CSV, AWS SCT crea tre file CSV.

Il primo file CSV contiene le informazioni seguenti relative alle azioni:

- Categoria
- Ricorrenza: il nome del file, il numero di riga e la posizione dell'elemento
- Numero dell'oggetto dell'azione
- Subject
- Group (Gruppo)

- Descrizione

- Riferimenti alla documentazione

- Azione consigliata

- Complessità stimata

Il secondo file CSV include il `Action_Items_Summary` suffisso nel nome e contiene le informazioni sul numero di ricorrenze di tutte le azioni.

Nell'esempio seguente, i valori nella colonna Sforzo della curva di apprendimento indicano la quantità di impegno necessaria per progettare un approccio alla conversione di ogni elemento di azione. I valori nella colonna Impegno per convertire un'occorrenza dell'azione indicano lo sforzo necessario per convertire ogni azione, seguendo l'approccio progettato. I valori utilizzati per indicare il livello di impegno necessario si basano su una scala ponderata, da basso (minimo) a elevato (massimo).

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

Il terzo file CSV include `Summary` nel suo nome e contiene il seguente riepilogo:

- Categoria

- Numero di oggetti

- Oggetti convertiti automaticamente

- Oggetti con azioni semplici

- Oggetti con azioni di media complessità

- Oggetti con azioni complesse

- Righe totali di codice

Creazione di un rapporto di valutazione multiserver per la migrazione del database

Per determinare l'obiettivo migliore per l'ambiente generale, crea un rapporto di valutazione multiserver.

Un rapporto di valutazione multiserver valuta più server in base all'input fornito per ogni definizione di schema che si desidera valutare. La definizione dello schema contiene i parametri di connessione al server del database e il nome completo di ogni schema. Dopo aver valutato ogni schema, AWS SCT produce un rapporto di valutazione riepilogativo e aggregato per la migrazione del database su più server. Questo rapporto mostra la complessità stimata per ogni possibile obiettivo di migrazione.

È possibile AWS SCT utilizzarlo per creare un rapporto di valutazione multiserver per i seguenti database di origine e destinazione.

Database di origine	Database di destinazione
Amazon Redshift	Amazon Redshift
Database SQL Azure	Aurora MySQL, MySQL, PostgreSQL
Azure Synapse Analytics	Amazon Redshift
BigQuery	Amazon Redshift
Greenplum	Amazon Redshift
IBM Db2 per z/OS	Amazon Aurora Edizione compatibile con MySQL (Aurora MySQL), Amazon Aurora edizione compatibile con PostgreSQL (Aurora PostgreSQL), MySQL, PostgreSQL, PostgreSQL
IBM Db2 LUW	Aurora MySQL, MariaDB, MySQL, PostgreSQL
Microsoft SQL Server	Aurora MySQL, Amazon Redshift, Babelfish per Aurora PostgreSQL, Microsoft SQL Server, MySQL, PostgreSQL, MySQL, PostgreSQL, MySQL, PostgreSQL, MySQL, PostgreSQL,

Database di origine	Database di destinazione
	MySQL, PostgreSQL, MySQL, PostgreSQL, MySQL, PostgreSQL, MySQL, PostgreSQL, MySQL, PostgreSQL
MySQL	Aurora MySQL, PostgreSQL, MySQL, PostgreSQL
Netezza	Amazon Redshift
Oracle	Aurora MySQL, Amazon Redshift, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL	Aurora MySQL, MySQL, PostgreSQL
SAP ASE	Aurora MySQL, MariaDB, MySQL, PostgreSQL
Snowflake	Amazon Redshift
Teradata	Amazon Redshift
Vertica	Amazon Redshift

Esecuzione di una valutazione multiserver

Utilizzare la seguente procedura per eseguire una valutazione multiserver con AWS SCT. Non è necessario creare un nuovo progetto AWS SCT per eseguire una valutazione multiserver. Prima di iniziare, assicurati di aver preparato un file CSV (Comma-Separated) con parametri di connessione al database. Inoltre, assicurati di aver installato tutti i driver del database richiesti e imposta la posizione dei driver nelle AWS SCT impostazioni. Per ulteriori informazioni, consulta [Scaricamento dei driver di database richiesti](#).

Per eseguire una valutazione multiserver e creare un rapporto di riepilogo aggregato

1. In AWS SCT, scegli File, Nuova valutazione multiserver. Viene visualizzata la finestra di dialogo Nuova valutazione multiserver.

New multiserver assessment ✕

Enter the project name, location to store reports and project files, and location of your connections file. ?

Project name

Location

Connections file

[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

2. Scegli Scarica un esempio di file di connessioni per scaricare un modello vuoto di un file CSV con parametri di connessione al database.
3. Immettete i valori per il nome del progetto, la posizione (per archiviare i report) e il file Connessioni (un file CSV).
4. Scegli CreaAWS SCT progetti per ogni database di origine per creare automaticamente progetti di migrazione dopo aver generato il rapporto di valutazione.
5. Con l'opzione CreaAWS SCT progetti per ogni database di origine attivata, puoi scegliere Aggiungi regole di mappatura a questi progetti e salvare le statistiche di conversione per l'uso offline. In questo caso, AWS SCT aggiungerà regole di mappatura a ciascun progetto e salverà i metadati del database di origine nel progetto. Per ulteriori informazioni, consulta [EsecuzioneAWS SCT in modalità offline](#).
6. Scegli Run (Esegui).

Viene visualizzata una barra di avanzamento che indica il ritmo della valutazione del database. Il numero di motori di destinazione può influire sul tempo di esecuzione della valutazione.

7. Scegliere Sì se viene visualizzato il seguente messaggio: L'analisi completa di tutti i server del database potrebbe richiedere del tempo. Vuoi procedere?

Al termine del rapporto di valutazione multiserver, viene visualizzata una schermata che lo indica.

8. Scegli Apri rapporto per visualizzare il rapporto di valutazione riassuntivo aggregato.

Per impostazione predefinita, AWS SCT genera un rapporto aggregato per tutti i database di origine e un rapporto di valutazione dettagliato per ogni nome di schema in un database di origine. Per ulteriori informazioni, consulta [Individuazione e visualizzazione dei report](#).

Con l'opzione Crea AWS SCT progetti per ogni database di origine attivata, AWS SCT crea un progetto vuoto per ogni database di origine. AWS SCT crea anche report di valutazione come descritto in precedenza. Dopo aver analizzato questi report di valutazione e aver scelto la destinazione della migrazione per ogni database di origine, aggiungi i database di destinazione a questi progetti vuoti.

Con l'opzione Aggiungi regole di mappatura a questi progetti e salva le statistiche di conversione per l'uso offline attivata, AWS SCT crea un progetto per ogni database di origine. Questi progetti includono le seguenti informazioni:

- Il tuo database di origine e una piattaforma di database di destinazione virtuale. Per ulteriori informazioni, consulta [Utilizzo di obiettivi virtuali](#).
- Una regola di mappatura per questa coppia sorgente-destinazione. Per ulteriori informazioni, consulta [Creazione di regole di mappatura](#).
- Un rapporto di valutazione della migrazione del database per questa coppia origine-destinazione.
- Metadati dello schema di origine, che consentono di utilizzare questo AWS SCT progetto in modalità offline. Per ulteriori informazioni, consulta [Esecuzione AWS SCT in modalità offline](#).

Preparazione di un file CSV di input

Per fornire parametri di connessione come input per il rapporto di valutazione multiserver, utilizza un file CSV come mostrato nell'esempio seguente.

```
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
```

L'esempio precedente utilizza un punto e virgola per separare i due nomi degli schemi per ilSales database. Utilizza inoltre un punto e virgola per separare le due piattaforme di migrazione del database di destinazione per ilSales database.

Inoltre, l'esempio precedente utilizzaAWS Secrets Manager per connettersi alCustomers database e l'autenticazione di Windows per connettersi alHR database.


Puoi creare un nuovo file CSV o scaricare un modello per un file CSVAWS SCT e inserire le informazioni richieste. Assicurati che la prima riga del file CSV includa gli stessi nomi di colonna mostrati nell'esempio precedente.

Per scaricare un modello del file CSV di input

1. Avvia AWS SCT.
2. Scegli File, quindi scegli Nuova valutazione multiserver.
3. Scegli Scarica un esempio di file di connessioni.

Assicurati che il tuo file CSV includa i seguenti valori, forniti dal modello:

- Nome: l'etichetta di testo che aiuta a identificare il database. AWS SCTvisualizza questa etichetta di testo nel rapporto di valutazione.
- Descrizione: un valore facoltativo, in cui è possibile fornire informazioni aggiuntive sul database.
- Secret Manager Key: il nome del segreto che memorizza le credenziali del database inAWS Secrets Manager. Per utilizzare Secrets Manager, assicurati di archiviareAWS i profili inAWS SCT. Per ulteriori informazioni, consulta [Utilizzo di AWS Secrets Manager](#).

 Important

AWS SCTignora il parametro Secret Manager Key se si includono i parametri Server IP, Port, Login e Password nel file di input.

- Server IP: il nome DNS (Domain Name Service) o l'indirizzo IP del server del database di origine.
- Porta: la porta utilizzata per connettersi al server del database di origine.
- Nome servizio: se si utilizza un nome di servizio per connettersi al database Oracle, il nome del servizio Oracle a cui connettersi.
- Nome del database: il nome del database. Per i database Oracle, utilizzare l'Oracle System ID (SID).

- BigQuery percorso: il percorso del file chiave dell'account di servizio per il BigQuery database di origine. Per ulteriori informazioni sulla creazione di questo file, consulta [Privilegi per BigQuery come fonte](#).
- Source Engine: il tipo di database di origine. Utilizza uno dei seguenti valori:
 - AZURE_MSSQL per un database SQL di Azure.
 - AZURE_SYNAPSE per un database Azure Synapse Analytics.
 - GOOGLE_BIGQUERY per un BigQuery database.
 - DB2ZOS per un database IBM Db2 for z/OS.
 - DB2LUW per un database IBM Db2 LUW.
 - GREENPLUM per un database Greenplum.
 - MSSQL per un database Microsoft SQL Server.
 - MYSQL per un database MySQL.
 - NETEZZA per un database Netezza.
 - ORACLE per un database Oracle.
 - POSTGRESQL per un database PostgreSQL.
 - REDSHIFT per un database Amazon Redshift.
 - SNOWFLAKE per un database Snowflake.
 - SYBASE_ASE per un database SAP ASE.
 - TERADATA per un database Teradata.
 - VERTICA per un database Vertica.
- Nomi degli schemi: i nomi degli schemi del database da includere nel rapporto di valutazione.

Per il database SQL di Azure, Azure Synapse Analytics BigQuery, Netezza, SAP ASE, Snowflake e SQL Server, usa il seguente formato del nome dello schema:

db_name.schema_name

Sostituire *db_name* con il nome del database di origine.

Sostituire *schema_name* con il nome dello schema di origine.

Racchiudere i nomi di database o schemi che includono un punto tra virgolette doppie come mostrato di seguito: "database.name". "schema.name".

Separare più nomi di schemi utilizzando il punto e virgola come mostrato di seguito:Schema1;Schema2.

I nomi del database e dello schema fanno distinzione tra maiuscole e minuscole.

Usa la percentuale (%) come jolly per sostituire qualsiasi numero di simboli nel database o nel nome dello schema. L'esempio precedente utilizza la percentuale (%) come jolly per includere tutti gli schemi delemployees database nel rapporto di valutazione.

- Usa l'autenticazione di Windows: se utilizzi l'autenticazione di Windows per connetterti al database Microsoft SQL Server, inserisci true. Per ulteriori informazioni, consulta [Utilizzo dell'autenticazione di Windows quando si utilizza Microsoft SQL Server come origine](#).
- Login: il nome utente per connettersi al server del database di origine.
- Password: la password per connettersi al server del database di origine.
- Usa SSL: se utilizzi Secure Sockets Layer (SSL) per connetterti al database di origine, inserisci true.
- Trust store: l'archivio di fiducia da utilizzare per la connessione SSL.
- Key store: l'archivio di chiavi da utilizzare per la connessione SSL.
- Autenticazione SSL: se utilizzi l'autenticazione SSL tramite certificato, inserisci true.
- Motori di destinazione: le piattaforme del database di destinazione. Utilizza i seguenti valori per specificare uno o più obiettivi nel rapporto di valutazione:
 - AURORA_MYSQL per un database Aurora compatibile con MySQL.
 - AURORA_POSTGRESQL per un database compatibile con Aurora PostgreSQL.
 - BABELFISH per un database Babelfish per Aurora PostgreSQL.
 - MARIA_DB per un database MariaDB.
 - MSSQL per un database Microsoft SQL Server.
 - MYSQL per un database MySQL.
 - ORACLE per un database Oracle.
 - POSTGRESQL per un database PostgreSQL.
 - REDSHIFT per un database Amazon Redshift.

Separa più obiettivi usando punti e virgola come questo:MYSQL;MARIA_DB. Il numero di obiettivi influisce sul tempo necessario per eseguire la valutazione.

Individuazione e visualizzazione dei report

La valutazione multiserver genera due tipi di report:

- Un rapporto aggregato di tutti i database di origine.
- Un rapporto di valutazione dettagliato dei database di destinazione per ogni nome di schema in un database di origine.

I report vengono archiviati nella directory scelta per Posizione nella finestra di dialogo di valutazione Nuovo multiserver.

Per accedere ai report dettagliati, è possibile navigare nelle sottodirectory, organizzate per database di origine, nome dello schema e motore del database di destinazione.

I report aggregati mostrano le informazioni in quattro colonne sulla complessità della conversione di un database di destinazione. Le colonne includono informazioni sulla conversione di oggetti di codice, oggetti di archiviazione, elementi di sintassi e complessità della conversione.

L'esempio seguente mostra le informazioni per la conversione di due schemi di database Oracle in Amazon RDS for PostgreSQL.

Server IP address and port	Secret Manager key	Name	Description	Database name	Schema name	Code object conversion % for "Amazon RDS for PostgreSQL"	Storage object conversion % for "Amazon RDS for PostgreSQL"	Syntax Elements conversion % for "Amazon RDS for PostgreSQL"	Conversion Complexity for "Amazon RDS for PostgreSQL"
192.0.2.0:1521		Sales		ORCL	Q4_2021	97.78%	100.00%	98.76%	1
192.0.2.0:1521		Sales		pdb	FY_2021	82.35%	85.19%	99.24%	10

Le stesse quattro colonne vengono aggiunte ai report per ogni motore di database di destinazione aggiuntivo specificato.

Per informazioni dettagliate su come leggere queste informazioni, consulta quanto segue.

Risultato di un rapporto di valutazione aggregato

Il rapporto di valutazione aggregato della migrazione del database multiserver in AWS Schema Conversion Tool è un file CSV con le seguenti colonne:

- 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*

Per raccogliere informazioni, AWS SCT esegue report di valutazione completi e quindi aggrega i report per schemi.

Nel rapporto, i tre campi seguenti mostrano la percentuale di possibile conversione automatica in base alla valutazione:

% di conversione dell'oggetto di codice

La percentuale di oggetti di codice nello schema che AWS SCT possono essere convertiti automaticamente o con modifiche minime. Gli elementi di codice includono procedure, funzioni, viste e simili.

% di conversione dell'oggetto di archiviazione

La percentuale di oggetti di archiviazione che SCT può convertire automaticamente o con modifiche minime. Gli oggetti di archiviazione includono tabelle, indici, vincoli e simili.

Conversione degli elementi di sintassi%

La percentuale di elementi di sintassi che SCT può convertire automaticamente. Gli elementi di sintassi includono `SELECT JOIN` clausole, e simili. `FROM DELETE`

Il calcolo della complessità della conversione si basa sulla nozione di elementi di azione. Un'azione riflette un tipo di problema riscontrato nel codice sorgente che è necessario risolvere manualmente durante la migrazione verso una determinata destinazione. Un'azione può avere più ricorrenze.

Una scala ponderata identifica il livello di complessità per eseguire una migrazione. Il numero 1 rappresenta il livello di complessità più basso e il numero 10 rappresenta il livello di complessità più elevato.

Conversione di schemi di database utilizzando AWS SCT

Puoi utilizzare AWS Schema Conversion Tool (AWS SCT) per convertire gli schemi del database esistenti da un motore di database a un altro. La conversione di un database tramite l'interfaccia AWS SCT utente può essere abbastanza semplice, ma ci sono diversi aspetti da considerare prima di eseguire la conversione.

Ad esempio, puoi usare AWS SCT per fare quanto segue:

- Puoi utilizzarlo AWS SCT per copiare uno schema di database locale esistente su un'istanza database Amazon RDS che esegue lo stesso motore. Puoi utilizzare questa caratteristica per analizzare il potenziale risparmio sui costi del passaggio al cloud e della modifica del tipo di licenza.
- In alcuni casi, le funzionalità del database non possono essere convertite in funzionalità Amazon RDS equivalenti. Se ospiti e gestisci automaticamente un database sulla piattaforma Amazon Elastic Compute Cloud (Amazon EC2), puoi emulare queste funzionalità AWS sostituendole con servizi.
- AWS SCT automatizza gran parte del processo di conversione dello schema del database di elaborazione delle transazioni online (OLTP) in un'istanza database MySQL di Amazon Relational Database Service (Amazon RDS), un cluster di database Amazon Aurora o un'istanza database PostgreSQL. I motori del database di origine e di destinazione contengono molte caratteristiche e funzionalità diverse e AWS SCT tentano di creare uno schema equivalente nell'istanza database di Amazon RDS laddove possibile. Se non è possibile alcuna conversione diretta, AWS SCT fornisce un elenco di operazioni possibili da svolgere.

Argomenti

- [Creazione di regole di migrazione in AWS SCT](#)
- [Conversione dello schema utilizzando AWS SCT](#)
- [Gestione delle conversioni manuali in AWS SCT](#)
- [Aggiornamento e aggiornamento dello schema convertito in AWS SCT](#)
- [Salvare e applicare lo schema convertito in AWS SCT](#)
- [Confronto degli schemi del database](#)
- [Ricerca di oggetti trasformati correlati](#)

AWS SCT supporta le seguenti conversioni OLTP (Online Transaction Processing).

Database di origine	Database di destinazione
IBM Db2 per z/OS (versione 12)	Edizione compatibile con Amazon Aurora MySQL, edizione compatibile con Amazon Aurora PostgreSQL, MySQL, PostgreSQL
IBM Db2 LUW (versioni 9.1, 9.5, 9.7, 10.5, 11.1 e 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Database SQL di Microsoft Azure	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
Microsoft SQL Server (versione 2008 R2 e successive)	Aurora MySQL, Aurora PostgreSQL, Babelfish per Aurora PostgreSQL, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL (versione 5.5 e successive)	Aurora PostgreSQL, MySQL, PostgreSQL È possibile migrare schemi e dati da MySQL a un cluster Aurora MySQL DB senza utilizzarli. AWS SCT Per ulteriori informazioni, consulta Migrazione dei dati su un cluster Amazon Aurora DB .
Oracle (versione 10.2 e successive)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL (versione 9.1 e successive)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
SAP ASE (12.5, 15.0, 15.5, 15.7 e 16.0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL

Per informazioni sulla conversione di uno schema di data warehouse, vedere [Conversione di schemi di data warehouse in Amazon Redshift utilizzando AWS SCT](#).

Per convertire lo schema del tuo database in Amazon RDS, segui i seguenti passaggi di alto livello:

- [Creazione di regole di migrazione in AWS SCT](#)— Prima di convertire lo schema con AWS SCT, puoi impostare regole che modificano il tipo di dati delle colonne, spostare oggetti da uno schema all'altro e modificare i nomi degli oggetti.
- [Conversione dello schema utilizzando AWS SCT](#)— AWS SCT crea una versione locale dello schema convertito da esaminare, ma non la applica all'istanza database di destinazione finché non sei pronto.
- [Creazione di report di valutazione della migrazione con AWS SCT](#)— AWS SCT crea un rapporto di valutazione della migrazione del database che descrive in dettaglio gli elementi dello schema che non possono essere convertiti automaticamente. Puoi utilizzare questo rapporto per identificare dove devi creare uno schema nella tua istanza database Amazon RDS compatibile con il tuo database di origine.
- [Gestione delle conversioni manuali in AWS SCT](#)— Se disponi di elementi dello schema che non possono essere convertiti automaticamente, hai due scelte: aggiornare lo schema di origine e riconvertirlo nuovamente oppure creare elementi dello schema equivalenti nell'istanza database Amazon RDS di destinazione.
- [Aggiornamento e aggiornamento dello schema convertito in AWS SCT](#)— Puoi aggiornare il tuo AWS SCT progetto con lo schema più recente dal tuo database di origine.
- [Salvare e applicare lo schema convertito in AWS SCT](#)— Quando sei pronto, AWS SCT applica lo schema convertito nel tuo progetto locale all'istanza database Amazon RDS di destinazione.

Creazione di regole di migrazione in AWS SCT

Prima di convertire lo schema con AWS SCT, puoi configurare le regole di migrazione. Le regole di migrazione in AWS SCT possono eseguire trasformazioni come modificare il tipo di dati delle colonne, spostare oggetti da uno schema all'altro e modificare i nomi degli oggetti. Ad esempio, supponiamo di avere un set di tabelle denominato `test_TABLE_NAME` nello schema di origine. È possibile impostare una regola che modifichi il prefisso con `test_` il prefisso `demo_` nello schema di destinazione.

Note

È possibile creare regole di migrazione solo per diversi motori di database di origine e di destinazione.

È possibile creare regole di migrazione che eseguano le seguenti attività:

- Aggiungere, rimuovere o sostituire un prefisso
- Aggiungere, rimuovere o sostituire un suffisso
- Cambia le regole di confronto delle colonne
- Modificare il tipo di dati
- Modificare la lunghezza di `char`, `varchar`, `nvarchar`, e `string` i tipi di dati
- Spostare oggetti
- Rinominare oggetti

È possibile creare regole di migrazione per i seguenti oggetti:

- Database
- Schema
- Tabella
- Colonna

Creazione di regole di migrazione

Puoi creare regole di migrazione e salvarle come parte del tuo progetto. Con il progetto aperto, utilizza la procedura seguente per creare regole di migrazione.

Per creare regole di migrazione

1. Nel menu **Visualizza**, selezionare **Visualizzazione mappatura**.
2. In **Mappature dei server**, scegli una coppia di server di origine e di destinazione.
3. Scegli **Nuova regola di migrazione**. Viene visualizzata la finestra di dialogo **Regole di trasformazione**.
4. Selezionare **Add new rule (Aggiungi nuova regola)**. Una nuova riga viene aggiunta all'elenco delle regole.
5. Configura la tua regola:
 - a. In **Name (Nome)** inserisci un nome per la regola.
 - b. In **Per**, scegliere il tipo di oggetto al quale verrà applicata la regola.
 - c. In **Per dove**, inserisci un filtro da applicare agli oggetti prima di applicare la regola di migrazione. La clausola `where` viene valutata utilizzando una clausola `like`. Puoi inserire

un nome esatto per selezionare un solo oggetto, oppure puoi inserire un modello per selezionare più oggetti.

I campi disponibili per la clausola where (dove) cambiano in base al tipo di oggetto. Ad esempio, se il tipo di oggetto è schema, esiste un solo campo disponibile per il nome schema.

- d. In Azioni, scegli il tipo di regola di migrazione che desideri creare.
- e. A seconda del tipo di regola, inserisci uno o due valori aggiuntivi. Ad esempio, per rinominare un oggetto, immettete il nuovo nome dell'oggetto. Per sostituire un prefisso, inserisci il vecchio prefisso e il nuovo prefisso.

Per i tipi di dati char, varchar, nvarchar e string, è possibile modificare la lunghezza del tipo di dati utilizzando l'operatore di moltiplicazione. Ad esempio, il `%*4` valore trasforma il tipo di `varchar(10)` dati in `varchar(40)`

6. Dopo aver configurato la regola di migrazione, scegli Salva per salvare la regola. Puoi inoltre selezionare Cancel (Annulla) per annullare le modifiche.

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. Dopo aver aggiunto, modificato ed eliminato le regole, scegliere Save All (Salva tutto) per salvare tutte le modifiche.
8. Scegliete Chiudi per chiudere la finestra di dialogo Regole di trasformazione.

Puoi utilizzare l'icona a forma di interruttore per disattivare una regola di migrazione senza eliminarla. È possibile utilizzare l'icona di copia per duplicare una regola di migrazione esistente. Puoi usare l'icona a forma di matita per modificare una regola di migrazione esistente. È possibile utilizzare l'icona di eliminazione per eliminare una regola di migrazione esistente. Per salvare le modifiche apportate alle regole di migrazione, scegli Salva tutto.

Esportazione delle norme sulla migrazione

Se si utilizza AWS DMS la migrazione dei dati dal database di origine al database di destinazione, è possibile fornire informazioni sulle regole di migrazione aAWS DMS. Per ulteriori informazioni sulle attività, vedere [Utilizzo delle attività di AWS Database Migration Service replica](#).

Per esportare le norme sulla migrazione

1. NelAWS Schema Conversion Tool, scegli Mapping View nel menu Visualizza.
2. In Regole di migrazione, scegli una regola di migrazione, quindi scegli Modifica regola di migrazione.
3. Scegli Esporta script per AWS DMS.
4. Sfogliare la posizione in cui si intende salvare lo script e scegliere Salva. Le regole di migrazione vengono salvate come script JSON che può essere utilizzato daAWS DMS.

Conversione dello schema utilizzando AWS SCT

Dopo aver collegato il progetto sia al database di origine che all'istanza database Amazon RDS di destinazione, il AWS Schema Conversion Tool progetto visualizza lo schema del database di origine nel pannello di sinistra. Lo schema si presenta in un formato di visualizzazione ad albero e ogni nodo della struttura è a caricamento differito. Quando scegli un nodo nella visualizzazione ad albero, AWS SCT richiede le informazioni sullo schema dal database di origine.

Puoi scegliere gli elementi dello schema dal database di origine e in seguito convertire lo schema in uno schema equivalente per il motore di database della tua istanza database di destinazione. Puoi scegliere qualsiasi elemento dello schema dal database di origine da convertire. Se l'elemento dello schema scelto dipende da un elemento padre, AWS SCT genera anche lo schema per detto elemento. Ad esempio, supponiamo di scegliere una tabella da convertire. In tal caso, AWS SCT genera lo schema per la tabella e il database in cui si trova la tabella.

Conversione dello schema

Per convertire uno schema dal database di origine, seleziona la casella di controllo relativa al nome dello schema da convertire. Quindi, scegli questo schema dal pannello di sinistra del tuo progetto. AWS SCT evidenzia il nome dello schema in blu. Apri il menu contestuale (clic con il pulsante destro del mouse) per lo schema e scegli Converti schema, come illustrato di seguito.

File Actions Main view Settings Applications Help Add source Add target

The screenshot shows the AWS Schema Conversion Tool interface. On the left, a tree view displays a list of servers and databases. The 'TEST' database is selected and highlighted in blue. A context menu is open over the 'TEST' database, with the 'Convert schema' option highlighted. The right pane shows the properties of the selected schema, including 'Created or last modified' (2021-09-06 09:56:08.26), 'Object name' (Name: TEST, compatibility-level: 100, collation-name: SQL_Latin1_General_CP1_CI_AS), and 'Category' (Name: <Aurora_MySQL (virtual)>).

Properties	
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	
Name	
Category	
Name	<Aurora_MySQL (virtual)>

Dopo aver convertito lo schema dal database di origine, puoi scegliere gli elementi dello schema dal pannello a sinistra del progetto e visualizzare lo schema convertito nei pannelli centrali del progetto. Il pannello centrale inferiore visualizza le proprietà e il comando SQL per creare lo schema convertito, come illustrato di seguito.

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the project structure, including a server named 'SQL Server - ec2-52-17-21-76.eu-west-1.cc' and a database named 'LARGE_DB_SS'. Under 'LARGE_DB_SS', there are two schemas: 'dbo' and 'Account'. The 'Account' table is selected. The main pane shows the SQL command for creating the table in the target Amazon RDS database. The command is as follows:

```

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,
13 )
14
15 ON [PRIMARY];

```

The interface also shows the 'Properties' tab for the table, which is currently empty. The cursor position is 0 and the table is modified.

Dopo aver convertito lo schema, puoi salvare il progetto. Le informazioni sullo schema del database di origine vengono salvate insieme al progetto. Questa funzionalità ti consente di lavorare offline senza essere connesso al database di origine. AWS SCT si connette al database di origine per aggiornare lo schema nel tuo progetto se scegli Refresh from Database (Aggiorna dal database) per il database di origine. Per ulteriori informazioni, consulta [Aggiornamento e aggiornamento dello schema convertito in AWS SCT](#).

Puoi creare un report di valutazione della migrazione del database degli elementi che non possono essere convertiti automaticamente. Il report di valutazione è utile per identificare e risolvere gli elementi dello schema che non possono essere convertiti automaticamente. Per ulteriori informazioni, consulta [Creazione di report di valutazione della migrazione con AWS SCT](#).

Quando AWS SCT genera uno schema convertito, non lo applica immediatamente all'istanza database di destinazione. Lo schema convertito rimane invece archiviato in locale fino a quando

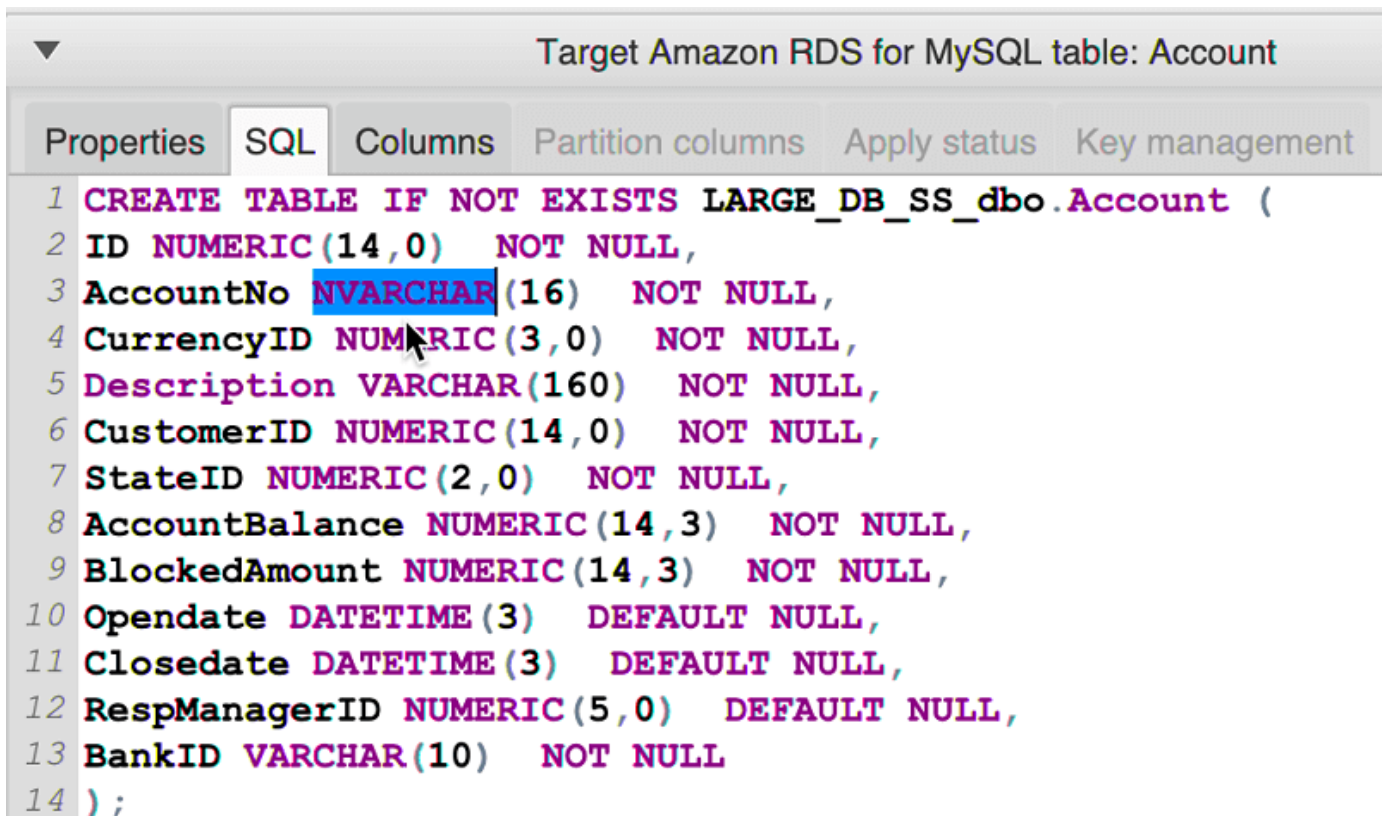
non sarai tu ad applicarlo all'istanza database di destinazione. Per ulteriori informazioni, consulta [Applicazione dello schema convertito](#).

Modifica dello schema convertito

Puoi modificare uno schema convertito e salvare le modifiche come parte del progetto.

Modifica di uno schema convertito

1. Nel pannello a sinistra, che visualizza lo schema del database di origine, scegliere l'elemento dello schema per il quale intendi modificare lo schema convertito.
2. Nel pannello centrale inferiore, che visualizza lo schema convertito dell'elemento selezionato, scegliere la scheda SQL.
3. Nel testo visualizzato nella scheda SQL, modificare lo schema a seconda delle necessità. Lo schema viene salvato automaticamente insieme al progetto quando lo aggiorni.



The screenshot shows a software interface with a title bar that reads "Target Amazon RDS for MySQL table: Account". Below the title bar is a tabbed interface with five tabs: "Properties", "SQL", "Columns", "Partition columns", "Apply status", and "Key management". The "SQL" tab is currently selected and active. The main area of the interface displays a SQL script for creating a table. The script 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 );
```

Le modifiche apportate allo schema convertito vengono memorizzate insieme al progetto mentre esegui gli aggiornamenti. Se hai convertito un nuovo elemento dello schema del database di origine, e hai eseguito degli aggiornamenti a uno schema convertito in precedenza per quell'elemento, gli

aggiornamenti esistenti vengono sostituiti dal nuovo elemento dello schema convertito in base al database di origine.

Cancellare uno schema convertito

Fino a quando non avrai applicato lo schema all'istanza database di destinazione, AWS SCT archivia lo schema convertito solamente in locale nel progetto. Puoi cancellare lo schema pianificato dal tuo progetto scegliendo il nodo di visualizzazione ad albero per l'istanza database e quindi scegliendo **Aggiorna dal database**. Poiché nessuno schema è stato scritto nell'istanza database di destinazione, l'aggiornamento dal database rimuove gli elementi dello schema pianificati nel AWS SCT progetto in modo che corrispondano a quelli esistenti nell'istanza database di origine.

Gestione delle conversioni manuali in AWS SCT

Il rapporto di valutazione include un elenco di elementi che non possono essere convertiti automaticamente nel motore di database dell'istanza database Amazon RDS di destinazione. Per ogni elemento che non può essere convertito, vi è un'azione nella scheda **Action Items (Attività)**.

Puoi rispondere alle attività nel report di valutazione nei seguenti modi:

- Modifica dello schema del database di origine.
- Modifica dello schema del database di destinazione.

Modifica dello schema di origine

Per alcuni elementi, potrebbe essere più semplice modificare lo schema del database nel database di origine in uno schema che può essere convertito automaticamente. In primo luogo, verifica che le nuove modifiche siano compatibili con l'architettura della tua applicazione, quindi aggiorna lo schema del database di origine. Infine, aggiorna il progetto con le informazioni sullo schema aggiornato. Ora puoi convertire lo schema aggiornato e generare un nuovo report di valutazione della migrazione del database. Le attività non saranno più visibili per gli elementi modificati nello schema di origine.

Il vantaggio di questo processo è che lo schema aggiornato è sempre disponibile quando esegui l'aggiornamento dal database di origine.

Modifica dello schema di destinazione

Per alcuni elementi, potrebbe essere più semplice applicare lo schema convertito al database di destinazione, e in seguito aggiungere manualmente gli elementi dello schema equivalente al

database di destinazione per gli elementi che non sono stati convertiti automaticamente. Puoi scrivere tutti gli schemi che possono essere convertiti automaticamente nell'istanza database di destinazione applicando lo schema. Per ulteriori informazioni, consulta [Salvare e applicare lo schema convertito in AWS SCT](#).

Gli schemi scritti nell'istanza database di destinazione non contengono gli elementi che non possono essere convertiti automaticamente. Dopo aver applicato lo schema all'istanza database di destinazione, puoi creare manualmente lo schema nell'istanza database di destinazione equivalente a quello del database di origine. Le attività del report di valutazione della migrazione del database contengono suggerimenti per la creazione dello schema equivalente.

Warning

Se crei manualmente lo schema nella tua istanza database di destinazione, salva una copia di qualsiasi tuo lavoro manuale. Se applichi di nuovo lo schema convertito dal tuo progetto all'istanza database di destinazione, il lavoro manuale viene sovrascritto.

In alcuni casi, potresti non essere in grado di creare uno schema equivalente nell'istanza database di destinazione. Potrebbe essere necessario riprogettare una porzione dell'applicazione e del database per utilizzare la funzionalità disponibile nel motore di database per l'istanza database di destinazione. In altri casi, puoi semplicemente ignorare lo schema che non può essere convertito automaticamente.

Aggiornamento e aggiornamento dello schema convertito in AWS SCT

Nel progetto AWS Schema Conversion Tool puoi aggiornare sia lo schema di origine che lo schema di destinazione.

- **Origine:** se si aggiorna lo schema per il database di origine, AWS SCT sostituisce lo schema del progetto con lo schema più recente del database di origine. Mediante questa funzionalità puoi aggiornare il progetto nel caso siano state apportate delle modifiche allo schema del database di origine.
- **Target:** se aggiorni lo schema per l'istanza database Amazon RDS di destinazione, AWS SCT sostituisce lo schema del progetto con lo schema più recente dell'istanza database di destinazione. Se non hai applicato lo schema all'istanza database di destinazione, AWS SCT cancella lo schema

convertito dal progetto. Quindi puoi convertire lo schema dal database di origine per una nuova istanza database di destinazione pulita.

Puoi aggiornare lo schema nel tuo AWS SCT progetto scegliendo **Aggiorna dal database**.

Note

Quando aggiorni lo schema, AWS SCT carica i metadati solo quando sono necessari. Per caricare completamente tutto lo schema del database, apri il menu contestuale (clic con il pulsante destro del mouse) dello schema e scegli **Carica schema**. Ad esempio, puoi utilizzare questa opzione per caricare tutti i metadati del tuo database contemporaneamente e quindi lavorare offline.

Salvare e applicare lo schema convertito in AWS SCT

Quando AWS Schema Conversion Tool genera lo schema convertito (come mostrato in [Conversione dello schema utilizzando AWS SCT](#)), lo schema convertito non viene applicato immediatamente all'istanza database di destinazione. Lo schema convertito rimane invece archiviato in locale nel progetto fino a quando non sarai tu ad applicarlo all'istanza database di destinazione. Con questa funzionalità puoi utilizzare gli elementi dello schema che non possono essere convertiti automaticamente nel motore del database di destinazione. Per ulteriori informazioni sugli elementi che non possono essere convertiti automaticamente, consulta [Creazione di report di valutazione della migrazione con AWS SCT](#).

Facoltativamente, puoi fare in modo che lo strumento salvi lo schema convertito in un file come script SQL prima di applicare lo schema all'istanza database di destinazione. Puoi anche fare in modo che lo strumento applichi lo schema convertito direttamente all'istanza database di destinazione.

Salvare lo schema convertito in un file

Puoi salvare lo schema convertito come script SQL in un file di testo. Utilizzando questo approccio, puoi modificare lo script SQL generato da AWS SCT per gestire gli elementi che lo strumento non è in grado di convertire automaticamente. Puoi quindi eseguire gli script aggiornati nell'istanza database di destinazione per applicare lo schema convertito al database di destinazione.

Per salvare lo schema convertito come script SQL

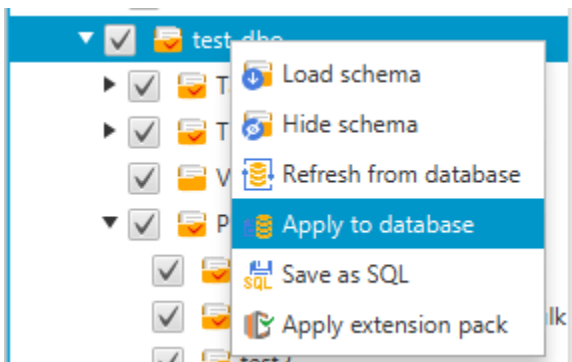
1. Scegli il tuo schema e apri il menu contestuale (clic con il pulsante destro del mouse).
2. Scegli Salva come SQL.
3. Inserisci il nome del file e scegli Salva.
4. Salva lo schema convertito utilizzando una delle seguenti opzioni:
 - File singolo
 - File singolo per fase
 - File singolo per dichiarazione

Per scegliere il formato dello script SQL

1. Nel menu Impostazioni, scegli Impostazioni progetto.
2. Scegli Salva script.
3. Per Vendor, scegli la piattaforma del database.
4. In Salva script SQL in, scegli come vuoi salvare lo script dello schema del database.
5. Scegliete OK per salvare le impostazioni.

Applicazione dello schema convertito

Quando sei pronto per applicare lo schema convertito all'istanza database Amazon RDS di destinazione, scegli l'elemento dello schema dal pannello di destra del tuo progetto. Apri il menu contestuale (tasto destro del mouse) per l'elemento dello schema, quindi scegli Apply to database (Applica al database), come illustrato di seguito.



Lo schema del pacchetto di estensione

Quando applichi per la prima volta lo schema convertito all'istanza database di destinazione, AWS SCT aggiunge un ulteriore schema all'istanza database di destinazione. Questo schema implementa le funzioni di sistema del database di origine richieste durante la scrittura dello schema convertito nell'istanza database di destinazione. Lo schema viene chiamato schema del pacchetto di estensione.

Non modificare lo schema del pacchetto di estensione, in quanto questo potrebbe comportare risultati inattesi nello schema convertito scritto nell'istanza database di destinazione. Quando lo schema è completamente migrato all'istanza database di destinazione, e AWS SCT non è più necessario, puoi eliminare lo schema del pacchetto di estensione.

Lo schema del pacchetto di estensione viene denominato in base al database di origine come segue:

- IBM Db2 LUW: `aws_db2_ext`
- Microsoft SQL Server: `aws_sqlserver_ext`
- MySQL: `aws_mysql_ext`
- Oracle: `aws_oracle_ext`
- PostgreSQL: `aws_postgresql_ext`
- CUSTODIA SAP: `aws_sapase_ext`

Per ulteriori informazioni, consulta [Utilizzo delle AWS Lambda funzioni del pacchetto di AWS SCT estensione](#).

Confronto degli schemi del database

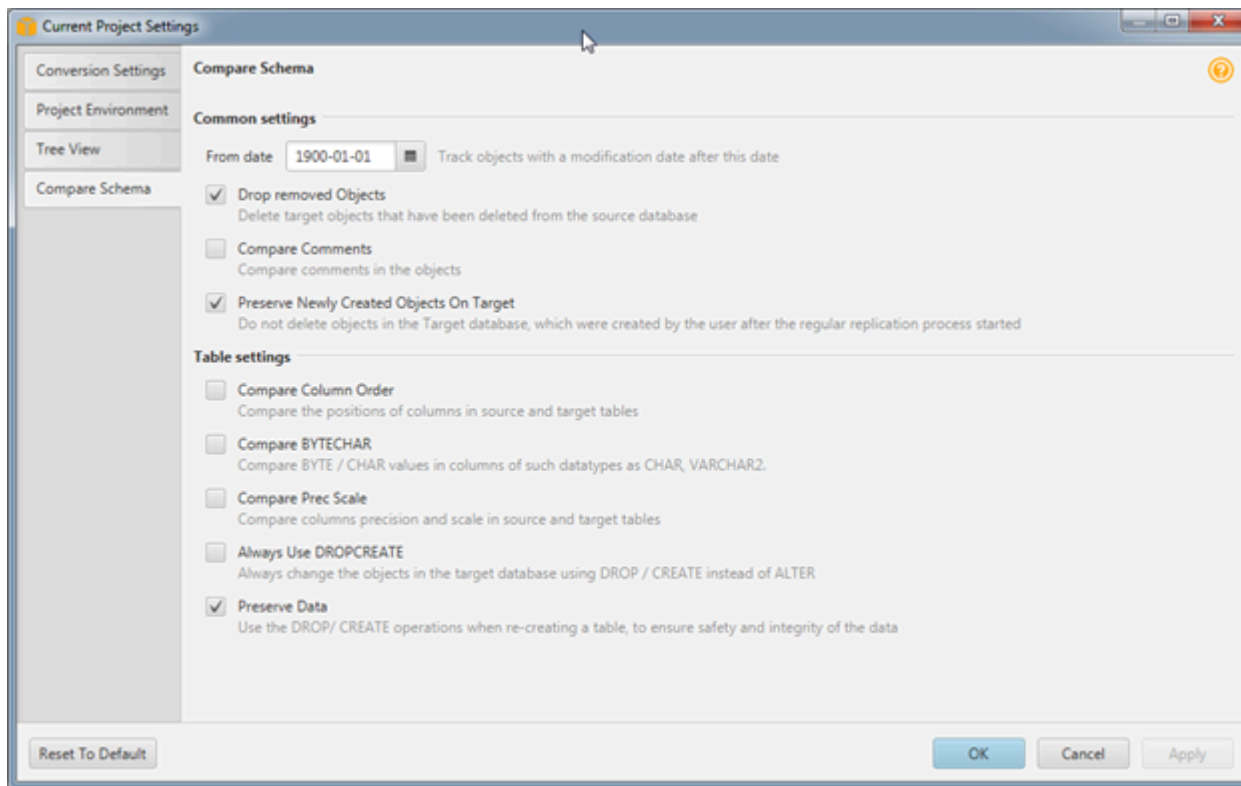
Se hai modificato lo schema di origine o di destinazione dopo la migrazione, puoi confrontare i due schemi del database utilizzando AWS SCT. Puoi confrontare gli schemi con versioni uguali o precedenti rispetto allo schema di origine.

Sono supportati i seguenti confronti dello schema:

- Da Oracle a Oracle, versioni 12.1.0.2.0, 11.1.0.7.0, 11.2.0.1.0 e 10
- Da SQL Server a SQL Server, versioni 2016, 2014, 2012, 2008 RD2, 2008
- Edizione compatibile con PostgreSQL e Aurora PostgreSQL, versioni 9.6, 9.5.9, 9.5.4

- Da MySQL a MySQL, versioni 5.6.36, 5.7.17, 5.5

Puoi specificare le impostazioni per il confronto degli schemi nella scheda Compare Schema (Confronta schema) della pagina Project Settings (Impostazioni progetto).

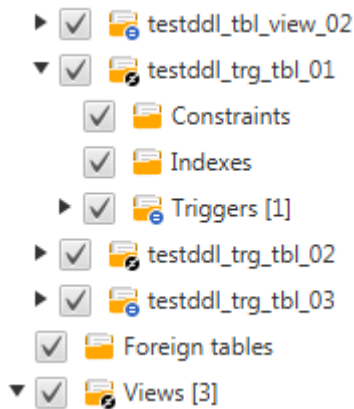


Per confrontare gli schemi, selezioni gli schemi e AWS SCT indica gli oggetti che differiscono tra i due schemi e gli oggetti che sono uguali.

Per confrontare due schemi

1. Aprire un progetto AWS SCT esistente oppure creare un progetto e connettere gli endpoint di origine e di destinazione.
2. Scegliere lo schema da confrontare.
3. Apri il menu contestuale (clic con il pulsante destro del mouse) e scegli Confronta schema.

AWS SCT indica gli oggetti diversi tra i due schemi aggiungendo un cerchio nero all'icona dell'oggetto.



Puoi applicare i risultati del confronto degli schemi rispetto a un singolo oggetto, a una singola categoria di oggetti o per l'intero schema. Scegli la casella accanto alla categoria, all'oggetto o allo schema che a cui vuoi applicare i risultati.

Ricerca di oggetti trasformati correlati

Dopo la conversione dello schema, in alcuni casi AWS SCT potrebbe aver creato diversi oggetti per un oggetto dello schema nel database di origine. Ad esempio, quando si esegue una conversione da Oracle a PostgreSQL, AWS SCT utilizza ogni trigger Oracle convertendoli in trigger e funzione di trigger nella destinazione PostgreSQL. Inoltre, quando AWS SCT converte una procedura o una funzione di pacchetto Oracle in PostgreSQL, viene creata una funzione equivalente e una funzione INIT che devono essere eseguite come init block prima che la procedura o la funzione possa essere eseguita.

La procedura seguente consente di visualizzare tutti gli oggetti correlati creati dopo la conversione dello schema.

Per visualizzare gli oggetti correlati creati durante la conversione dello schema

1. Dopo la conversione dello schema, scegli l'oggetto convertito nella visualizzazione ad albero di destinazione.
2. Scegliere la scheda Related Converted Objects (Oggetti convertiti correlati).
3. Visualizza l'elenco degli oggetti di destinazione correlati.

Conversione di schemi di data warehouse in Amazon Redshift utilizzando AWS SCT

The AWS Schema Conversion Tool (AWS SCT) automatizza gran parte del processo di conversione dello schema del data warehouse in uno schema di database Amazon Redshift. Poiché i motori di database di origine e destinazione possono avere molte caratteristiche e funzionalità diverse, se possibile AWS SCT tenta di creare uno schema equivalente nel database di destinazione. Se non è possibile alcuna conversione diretta, AWS SCT fornisce un report di valutazione contenente un elenco di operazioni possibili da svolgere. Utilizzando AWS SCT, puoi gestire le chiavi, mappare i tipi di dati e di oggetti e creare conversioni manuali.

AWS SCT può convertire i seguenti schemi di data warehouse in Amazon Redshift.

- Amazon Redshift
- Azure Synapse Analytics (versione 10)
- BigQuery
- Greenplum Database (versione 4.3)
- Microsoft SQL Server (versione 2008 e successive)
- Netezza (versione 7.0.3 e successive)
- Oracle (versione 10.2 e successive)
- Snowflake (versione 3)
- Teradata (versione 13 e successive)
- Vertica (versione 7.2 e successive)

Per informazioni sulla conversione di uno schema di database di elaborazione delle transazioni online (OLTP), vedere. [Conversione di schemi di database utilizzando AWS SCT](#)

Per convertire uno schema di data warehouse, procedi nel seguente modo:

1. Specifica la strategia e le regole di ottimizzazione e specifica le regole di migrazione che desideri AWS SCT utilizzare. Puoi impostare regole che consentono di modificare i tipi di dati delle colonne, spostare gli oggetti da uno schema a un altro e modificare i nomi degli oggetti.

Puoi specificare le regole di ottimizzazione e migrazione in Impostazioni. Per ulteriori informazioni sulle strategie di ottimizzazione, vedere [Scelta delle strategie e delle regole di ottimizzazione da](#)

- [utilizzare con AWS SCT](#). per ulteriori informazioni sulle regole di migrazione, vedere [Creazione di regole di migrazione in AWS SCT](#)
2. Fornisci le statistiche sul tuo data warehouse di origine affinché AWS SCT possa ottimizzare il modo in cui il data warehouse viene convertito. Puoi raccogliere le statistiche direttamente dal database oppure caricare un file di statistiche esistente. Per ulteriori informazioni su come fornire le statistiche sul data warehouse, consulta [Raccolta o caricamento di statistiche per AWS SCT](#).
 3. Crea un report di valutazione della migrazione del database che elenchi gli elementi dello schema che non possono essere convertiti automaticamente. Puoi usare questo report per individuare dove è necessario creare manualmente uno schema nel database di destinazione che sia compatibile con il database di origine. Per ulteriori informazioni sul report di valutazione, consulta [Creazione di report di valutazione della migrazione con AWS SCT](#).
 4. Converti lo schema: AWS SCT crea una versione locale dello schema convertito perché tu lo possa verificare, ma non lo applica al database di destinazione fino a quando non sei pronto. Per ulteriori informazioni sulla conversione, consulta [Convertire lo schema utilizzando AWS SCT](#).
 5. Dopo aver convertito lo schema, puoi gestire e modificare le chiavi. La gestione delle chiavi rappresenta la base della conversione del data warehouse. Per ulteriori informazioni sulla gestione delle chiavi, consulta [Gestione e personalizzazione delle chiavi in AWS SCT](#).
 6. Se disponi di elementi dello schema che non possono essere convertiti automaticamente, puoi decidere di aggiornare lo schema e convertirlo nuovamente oppure di creare elementi dello schema equivalenti nel database di destinazione. Per ulteriori informazioni sulla conversione degli elementi dello schema, consulta [Gestione delle conversioni manuali in AWS SCT](#). Per ulteriori informazioni sull'aggiornamento dello schema di origine, consulta [Aggiornamento e aggiornamento dello schema convertito in AWS SCT](#).
 7. Quando sei pronto, puoi applicare lo schema convertito al database di destinazione. Per ulteriori informazioni sul salvataggio e l'applicazione dello schema convertito, consulta [Salvare e applicare lo schema convertito in AWS SCT](#).

Autorizzazioni per Amazon Redshift come obiettivo

Le autorizzazioni richieste per Amazon Redshift come destinazione sono elencate di seguito:

- CREATE ON DATABASE: consente di creare nuovi schemi nel database.
- CREATE ON SCHEMA: consente di creare oggetti nello schema del database.
- GRANT USAGE ON LANGUAGE: consente di creare nuove funzioni e procedure nel database.

- GRANT SELECT SU TUTTE LE TABELLE NELLO SCHEMA `pg_catalog`: fornisce all'utente informazioni di sistema sul cluster Amazon Redshift.
- GRANT SELECT ON `pg_class_info`: fornisce all'utente informazioni sullo stile di distribuzione delle tabelle.

È possibile utilizzare il seguente esempio di codice per creare un utente del database e concedere le autorizzazioni.

```
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;
```

Nell'esempio precedente, sostituisci *user_name* con il nome del tuo utente. Quindi, sostituisci *db_name* con il nome del tuo database Amazon Redshift di destinazione. Quindi, sostituisci *schema_name* con il nome del tuo schema Amazon Redshift. Ripeti l'GRANT CREATE ON SCHEMA operazione per ogni schema di destinazione in cui applicherai il codice convertito o migrerai i dati. Infine, sostituisci *your_password* con una password sicura.

Puoi applicare un pacchetto di estensione al database Amazon Redshift di destinazione. Un pacchetto di estensione è un modulo aggiuntivo che emula le funzioni del database di origine necessarie per la conversione di oggetti in Amazon Redshift. Per ulteriori informazioni, consulta [Utilizzo dei pacchetti di AWS SCT estensione](#).

Per questa operazione, è AWS SCT necessaria l'autorizzazione per accedere al bucket Amazon S3 per tuo conto. Per fornire questa autorizzazione, crea un utente AWS Identity and Access Management (IAM) con la seguente politica.

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

```
    "Action":[
      "s3:DeleteObject",
      "s3:GetObject",
      "s3:ListBucket",
      "s3:PutObject"
    ],
    "Resource":[
      "arn:aws:s3:::aws-sct-*"
    ]
  },
  {
    "Effect":"Allow",
    "Action":[
      "s3:ListAllMyBuckets"
    ],
    "Resource":""
  }
]
```

Scelta delle strategie e delle regole di ottimizzazione da utilizzare con AWS SCT

Per ottimizzare le modalità utilizzata da AWS Schema Conversion Tool per convertire lo schema del data warehouse, puoi scegliere le strategie e le regole che desideri vengano utilizzate dallo strumento. Dopo aver convertito lo schema e verificato le chiavi suggerite, puoi adeguare le regole o modificare la tua strategia per ottenere i risultati desiderati.

Scelta delle strategie di ottimizzazione e delle regole

1. Scegli Settings (Impostazioni), quindi seleziona Project Settings (Impostazioni di progetto). Verrà visualizzata la finestra di dialogo Current project settings (Impostazioni di progetto correnti).
2. Nel riquadro a sinistra scegli Optimization Strategies (Strategie di ottimizzazione). Le strategie di ottimizzazione appariranno nel riquadro a destra con le impostazioni predefinite selezionate.
3. In Strategy Sector (Settore delle strategie), scegli la strategia di ottimizzazione che intendi utilizzare. Puoi scegliere tra le seguenti opzioni:
 - Usa i metadati, ignora le informazioni statistiche: in questa strategia, solo le informazioni dei metadati vengono utilizzate per le decisioni di ottimizzazione. Ad esempio, se in una tabella di

origine è presente più di un indice, viene utilizzato l'ordine del database di origine, e il primo indice diventa una chiave di distribuzione.

- Ignora i metadati, utilizza le informazioni statistiche: in questa strategia, le decisioni di ottimizzazione derivano solo da informazioni statistiche. Questa strategia si applica solo alle tabelle e alle colonne per le quali vengono fornite le statistiche. Per ulteriori informazioni, consulta [Raccolta o caricamento di statistiche per AWS SCT](#).
 - Usa i metadati e utilizza le informazioni statistiche: in questa strategia, sia i metadati che le statistiche vengono utilizzati per le decisioni di ottimizzazione.
4. Dopo aver scelto la strategia di ottimizzazione puoi scegliere le regole che intendi utilizzare. Puoi scegliere tra le seguenti opzioni:
- Choose Distribution Key and Sort Keys using metadata (Scegli la chiave di distribuzione e le chiavi di ordinamento tramite i metadati)
 - Choose fact table and appropriate dimension for collation (Scegli le tabelle dei fatti e la dimensione adatta per le regole di confronto)
 - Analyze cardinality of indexes' columns (Analizza la cardinalità delle colonne di indici)
 - Trova le tabelle e le colonne più utilizzate dalla tabella del registro delle query

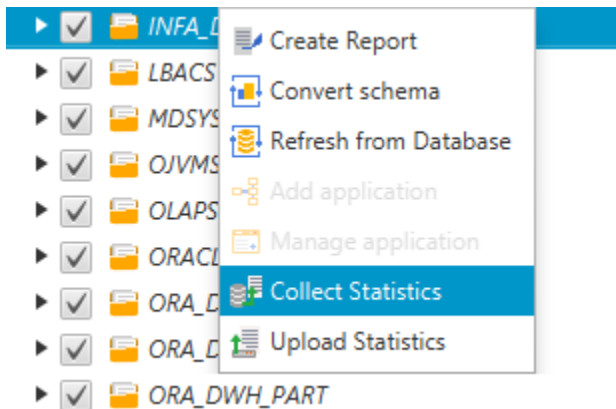
Per ogni regola, puoi inserire un valore per la chiave di ordinamento e un peso per la chiave di distribuzione. AWS SCT usa il peso selezionato quando converte il tuo schema. In seguito, dopo aver verificato le chiavi suggerite, se non sei soddisfatto dei risultati puoi tornare a questa fase e modificare le impostazioni. Per ulteriori informazioni, consulta [Gestione e personalizzazione delle chiavi in AWS SCT](#).

Raccolta o caricamento di statistiche per AWS SCT

Per ottimizzare le modalità utilizzata da AWS Schema Conversion Tool per convertire lo schema del data warehouse, puoi fornire le statistiche dal database di origine utilizzabili dallo strumento. Puoi raccogliere le statistiche direttamente dal database oppure caricare un file di statistiche esistente.

Fornire e verificare le statistiche

1. Apri il tuo progetto ed effettua la connessione al database di origine.
2. Scegli un oggetto dello schema dal pannello a sinistra del progetto e apri il menu contestuale (tasto destro del mouse) per l'oggetto. Scegli Collect Statistics (Raccogli statistiche) o Upload Statistics (Carica statistiche) come illustrato di seguito.



3. Scegli un oggetto dello schema dal pannello a sinistra del progetto e scegli la scheda Statistics (Statistiche). Puoi verificare le statistiche dell'oggetto.

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

In seguito, dopo aver verificato le chiavi suggerite, se non sei soddisfatto dei risultati puoi raccogliere statistiche aggiuntive e ripetere questa procedura. Per ulteriori informazioni, consulta [Gestione e personalizzazione delle chiavi in AWS SCT](#).

Creazione di regole di migrazione in AWS SCT

Prima di convertire lo schema con AWS SCT, puoi configurare le regole di migrazione. Le regole di migrazione possono fare cose come modificare il tipo di dati delle colonne, spostare oggetti da uno schema all'altro e modificare i nomi degli oggetti. Ad esempio, supponiamo di avere un set di tabelle denominato `test_TABLE_NAME` nello schema di origine. È possibile impostare una regola che modifichi il prefisso con `test_` il prefisso `demo_` nello schema di destinazione.

Note

È possibile creare regole di migrazione solo per diversi motori di database di origine e di destinazione.

È possibile creare regole di migrazione che eseguano le seguenti attività:

- Aggiungere, rimuovere o sostituire un prefisso
- Aggiungere, rimuovere o sostituire un suffisso
- Cambia le regole di confronto delle colonne
- Modificare il tipo di dati
- Modificare la lunghezza di `char`, `varchar`, `nvarchar`, e `string` i tipi di dati
- Spostare oggetti
- Rinominare oggetti

È possibile creare regole di migrazione per i seguenti oggetti:

- Database
- Schema
- Tabella
- Colonna

Creazione di regole di migrazione

Puoi creare regole di migrazione e salvarle come parte del tuo progetto. Con il progetto aperto, utilizza la procedura seguente per creare regole di migrazione.

Per creare regole di migrazione

1. Nel menu *Visualizza*, selezionare *Visualizzazione mappatura*.
2. In *Mappature dei server*, scegli una coppia di server di origine e di destinazione.
3. Scegli *Nuova regola di migrazione*. Viene visualizzata la finestra di dialogo *Regole di trasformazione*.
4. Selezionare *Add new rule (Aggiungi nuova regola)*. Una nuova riga viene aggiunta all'elenco delle regole.
5. Configura la tua regola:
 - a. In *Name (Nome)* inserisci un nome per la regola.
 - b. In *Per*, scegliere il tipo di oggetto al quale verrà applicata la regola.
 - c. Per *dove*, inserisci un filtro da applicare agli oggetti prima di applicare la regola di migrazione. La clausola *where* viene valutata utilizzando una clausola *like*. Puoi inserire un nome esatto per selezionare un solo oggetto, oppure puoi inserire un modello per selezionare più oggetti.

I campi disponibili per la clausola *where* (*dove*) cambiano in base al tipo di oggetto. Ad esempio, se il tipo di oggetto è *schema*, esiste un solo campo disponibile per il nome *schema*.
 - d. In *Azioni*, scegli il tipo di regola di migrazione che desideri creare.
 - e. A seconda del tipo di regola, inserisci uno o due valori aggiuntivi. Ad esempio, per rinominare un oggetto, immettete il nuovo nome dell'oggetto. Per sostituire un prefisso, inserisci il vecchio prefisso e il nuovo prefisso.
6. Dopo aver configurato la regola di migrazione, scegli *Salva* per salvare la regola. Puoi inoltre selezionare *Cancel (Annulla)* per annullare le modifiche.

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. Dopo aver aggiunto, modificato ed eliminato le regole, scegliere Save All (Salva tutto) per salvare tutte le modifiche.
8. Scegliete Chiudi per chiudere la finestra di dialogo Regole di trasformazione.

Puoi utilizzare l'icona a forma di interruttore per disattivare una regola di migrazione senza eliminarla. È possibile utilizzare l'icona di copia per duplicare una regola di migrazione esistente. Puoi usare l'icona a forma di matita per modificare una regola di migrazione esistente. È possibile utilizzare l'icona di eliminazione per eliminare una regola di migrazione esistente. Per salvare le modifiche apportate alle regole di migrazione, scegli Salva tutto.

Esportazione delle norme sulla migrazione

Se usi AWS Database Migration Service (AWS DMS) per migrare i tuoi dati dal tuo database di origine al database di destinazione, puoi fornire informazioni sulle tue regole di migrazione a AWS DMS. Per ulteriori informazioni sulle attività, vedere [Utilizzo delle attività di AWS Database Migration Service replica](#).

Per esportare le norme sulla migrazione

1. Nel AWS Schema Conversion Tool, scegli Mapping View nel menu Visualizza.
2. In Regole di migrazione, scegli una regola di migrazione, quindi scegli Modifica regola di migrazione.

3. Scegli Esporta script per AWS DMS.
4. Sfogliare la posizione in cui si intende salvare lo script e scegliere Salva. Le regole di migrazione vengono salvate come script JSON che può essere utilizzato da AWS DMS.

Convertire lo schema utilizzando AWS SCT

Dopo aver connesso il progetto sia al database di origine che al database di destinazione, il progetto AWS Schema Conversion Tool visualizza lo schema dal database di origine nel riquadro sinistra. Lo schema si presenta in un formato di visualizzazione ad albero e ogni nodo della struttura è a caricamento differito. Quando scegli un nodo nella visualizzazione ad albero, AWS SCT richiede le informazioni sullo schema dal database di origine.

Puoi scegliere gli elementi dello schema dal database di origine, e in seguito convertire lo schema in uno schema equivalente per il motore di database del database di destinazione. Puoi scegliere qualsiasi elemento dello schema dal database di origine da convertire. Se l'elemento dello schema scelto dipende da un elemento padre, AWS SCT genera anche lo schema per detto elemento. Ad esempio, se scegli una colonna da una tabella da convertire, AWS SCT genera lo schema per la colonna, per la tabella in cui si trova detta colonna e per il database in cui si trova la tabella.

Conversione dello schema

Per convertire uno schema dal database di origine, seleziona la casella di controllo relativa al nome dello schema da convertire. Quindi, scegli questo schema dal pannello di sinistra del tuo progetto. AWS SCT evidenzia il nome dello schema in blu. Apri il menu contestuale (clic con il pulsante destro del mouse) per lo schema e scegli Converti schema, come illustrato di seguito.

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)>

Dopo aver convertito lo schema dal database di origine, puoi scegliere gli elementi dello schema dal pannello a sinistra del progetto e visualizzare lo schema convertito nei pannelli centrali del progetto. Il pannello centrale inferiore visualizza le proprietà e il comando SQL per creare lo schema convertito, come illustrato di seguito.

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]' > 'LARGE_DB_SS' > 'Schemas [2]' > 'dbo' > 'Tables [4]' > 'Account'. The main area shows the SQL editor with two tabs: 'SQL' and 'Columns'. The 'SQL' tab displays the original MySQL table definition and its converted SQL Server equivalent.

```

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];

```

Cursor position: 0 Modified: true

Target Amazon RDS for MySQL table: Account

```

1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo VARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,

```

Dopo aver convertito lo schema, puoi salvare il progetto. Le informazioni sullo schema del database di origine vengono salvate insieme al progetto. Questa funzionalità ti consente di lavorare offline senza essere connesso al database di origine. AWS SCT si connette al database di origine per aggiornare lo schema nel tuo progetto se scegli Refresh from Database (Aggiorna dal database) per il database di origine. Per ulteriori informazioni, consulta [Aggiornamento e aggiornamento dello schema convertito in AWS SCT](#).

Puoi creare un report di valutazione della migrazione del database degli elementi che non possono essere convertiti automaticamente. Il report di valutazione è utile per identificare e risolvere gli elementi dello schema che non possono essere convertiti automaticamente. Per ulteriori informazioni, consulta [Creazione di report di valutazione della migrazione con AWS SCT](#).

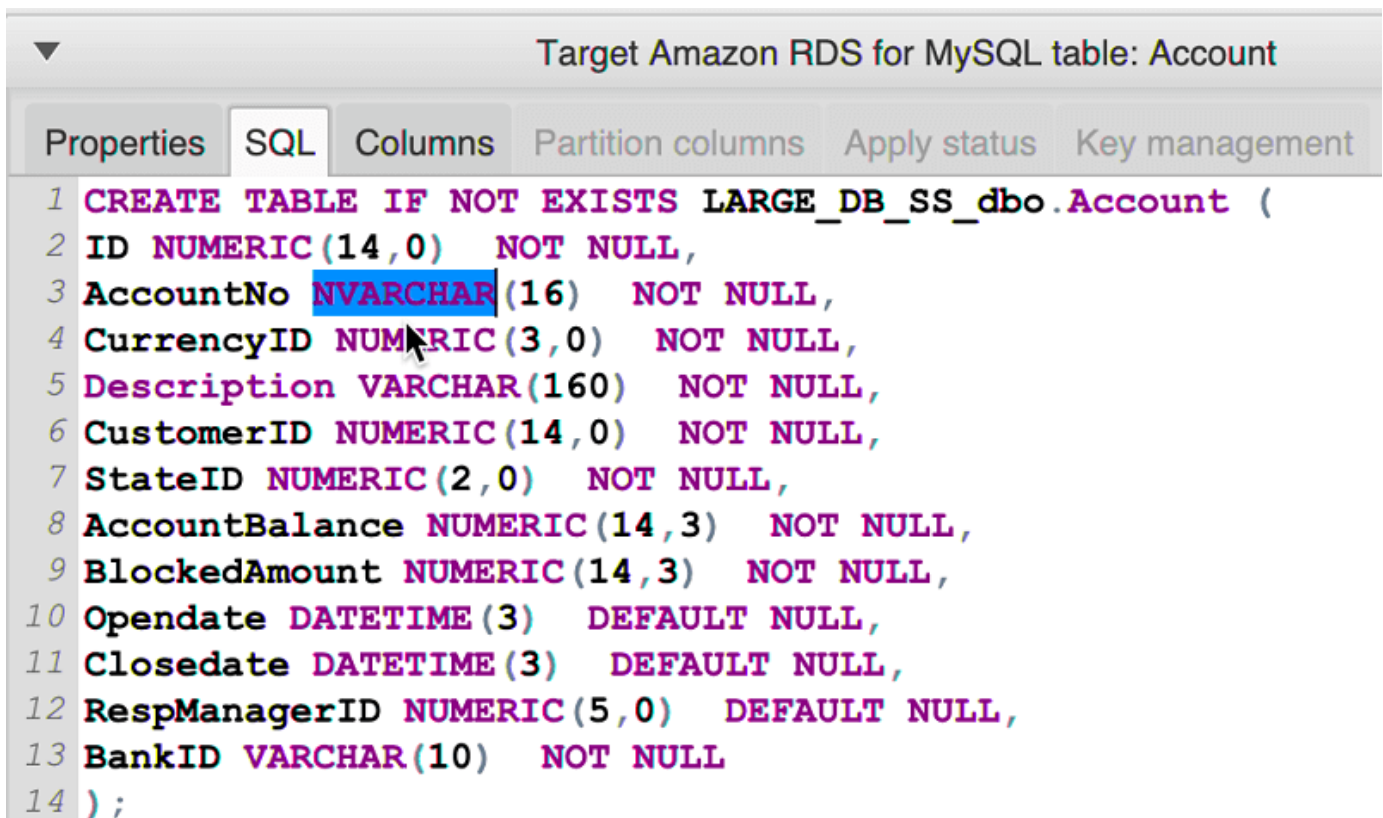
Quando AWS SCT genera uno schema convertito, non lo applica immediatamente al database di destinazione. Lo schema convertito rimane invece archiviato localmente fino a quando non sarai tu ad applicarlo al database di destinazione. Per ulteriori informazioni, consulta [Applicazione dello schema convertito](#).

Modifica dello schema convertito

Puoi modificare uno schema convertito e salvare le modifiche come parte del progetto.

Modifica di uno schema convertito

1. Nel pannello a sinistra, che visualizza lo schema del database di origine, scegliere l'elemento dello schema per il quale intendi modificare lo schema convertito.
2. Nel pannello centrale inferiore, che visualizza lo schema convertito dell'elemento selezionato, scegliere la scheda SQL.
3. Nel testo visualizzato nella scheda SQL, modificare lo schema a seconda delle necessità. Lo schema viene salvato automaticamente insieme al progetto quando lo aggiorni.



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 text 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 );
```

Le modifiche apportate allo schema convertito vengono memorizzate insieme al progetto mentre esegui gli aggiornamenti. Se hai convertito un nuovo elemento dello schema del database di origine, e hai eseguito degli aggiornamenti a uno schema convertito in precedenza per quell'elemento, gli aggiornamenti esistenti vengono sostituiti dal nuovo elemento dello schema convertito in base al database di origine.

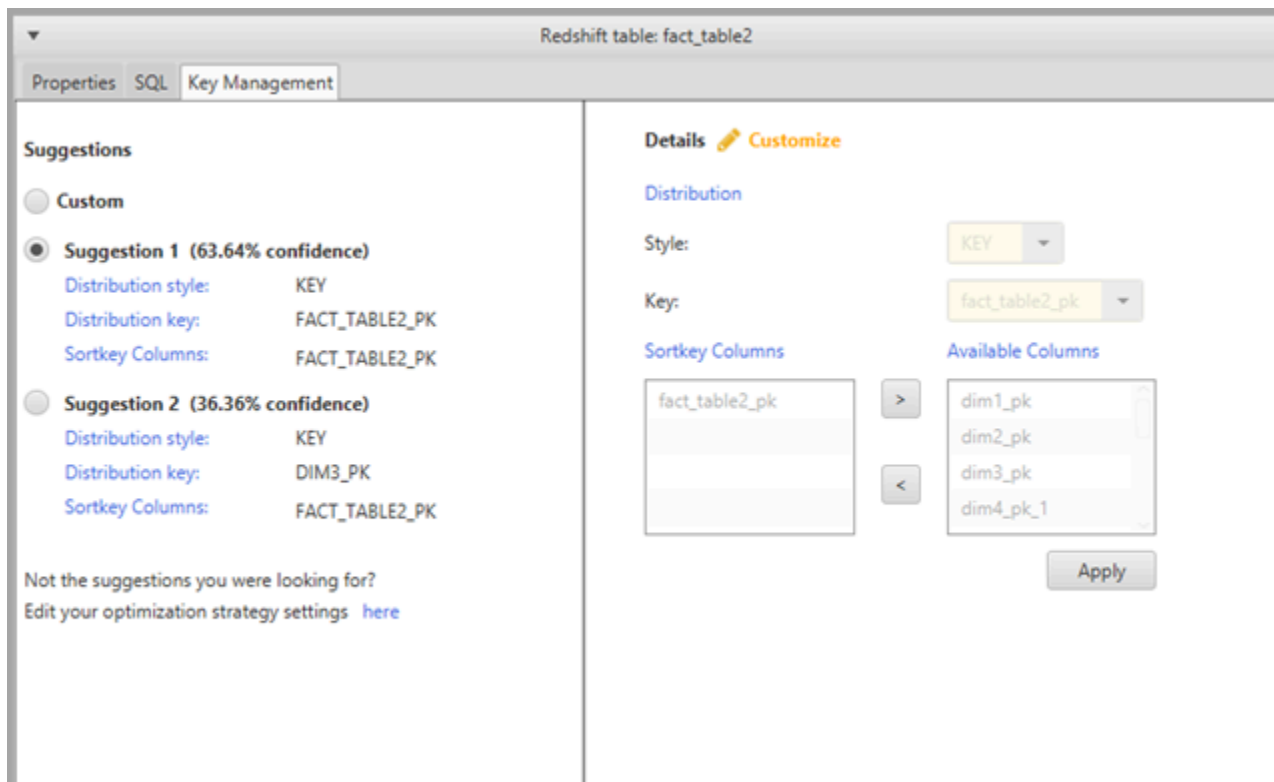
Cancellare uno schema convertito

Fino a quando non avrai applicato lo schema al database di destinazione, AWS SCT archivia lo schema convertito solo localmente nel progetto. Puoi eliminare lo schema pianificato dal progetto scegliendo il nodo della visualizzazione ad albero per il database di destinazione e selezionando in seguito Refresh from Database (Aggiorna dal database). Poiché non è stato scritto alcuno schema nel database di destinazione, l'aggiornamento dal database rimuove gli elementi dello schema pianificato nel progetto AWS SCT per adattarsi a ciò che è presente nel database di destinazione.

Gestione e personalizzazione delle chiavi in AWS SCT

Dopo aver convertito lo schema con AWS Schema Conversion Tool, puoi gestire e modificare le chiavi. La gestione delle chiavi rappresenta la base della conversione del data warehouse.

Per la gestione delle chiavi, seleziona una tabella nel database di destinazione e scegli la scheda Key Management (Gestione delle chiavi) come illustrato di seguito.



Il riquadro a sinistra contiene i suggerimenti per le chiavi e include il grado di affidabilità per ogni suggerimento. Puoi scegliere uno dei suggerimenti o personalizzare la chiave modificandola nel riquadro a destra.

Se le opzioni della chiave non hanno l'aspetto sperato, puoi modificare le strategie di ottimizzazione e riprovare la conversione. Per ulteriori informazioni, consulta [Scelta delle strategie e delle regole di ottimizzazione da utilizzare con AWS SCT](#).

Argomenti correlati

- [Scegli la chiave di ordinamento migliore](#)
- [Scegli lo stile di distribuzione migliore](#)

Creazione e utilizzo del rapporto di valutazione in AWS SCT

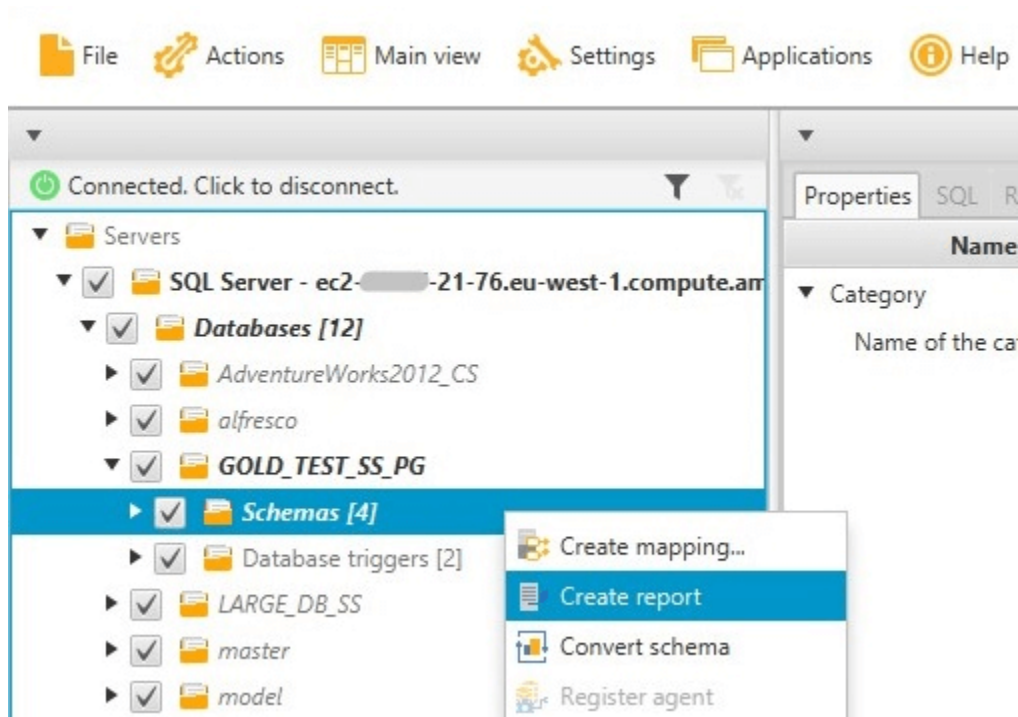
AWS Schema Conversion Tool crea un report di valutazione della migrazione del database per aiutarti a convertire lo schema. Il rapporto di valutazione della migrazione del database fornisce informazioni importanti sulla conversione dello schema dal database di origine al database di destinazione. Il rapporto riassume tutte le attività di conversione dello schema e descrive in dettaglio le azioni per lo schema che non possono essere convertite nel motore DB del database di destinazione. Il report include inoltre le stime dello sforzo necessario per scrivere nel database di destinazione il codice equivalente che non può essere convertito automaticamente.

Creazione di un rapporto di valutazione della migrazione del database

Utilizza la procedura seguente per creare un report di valutazione della migrazione del database.

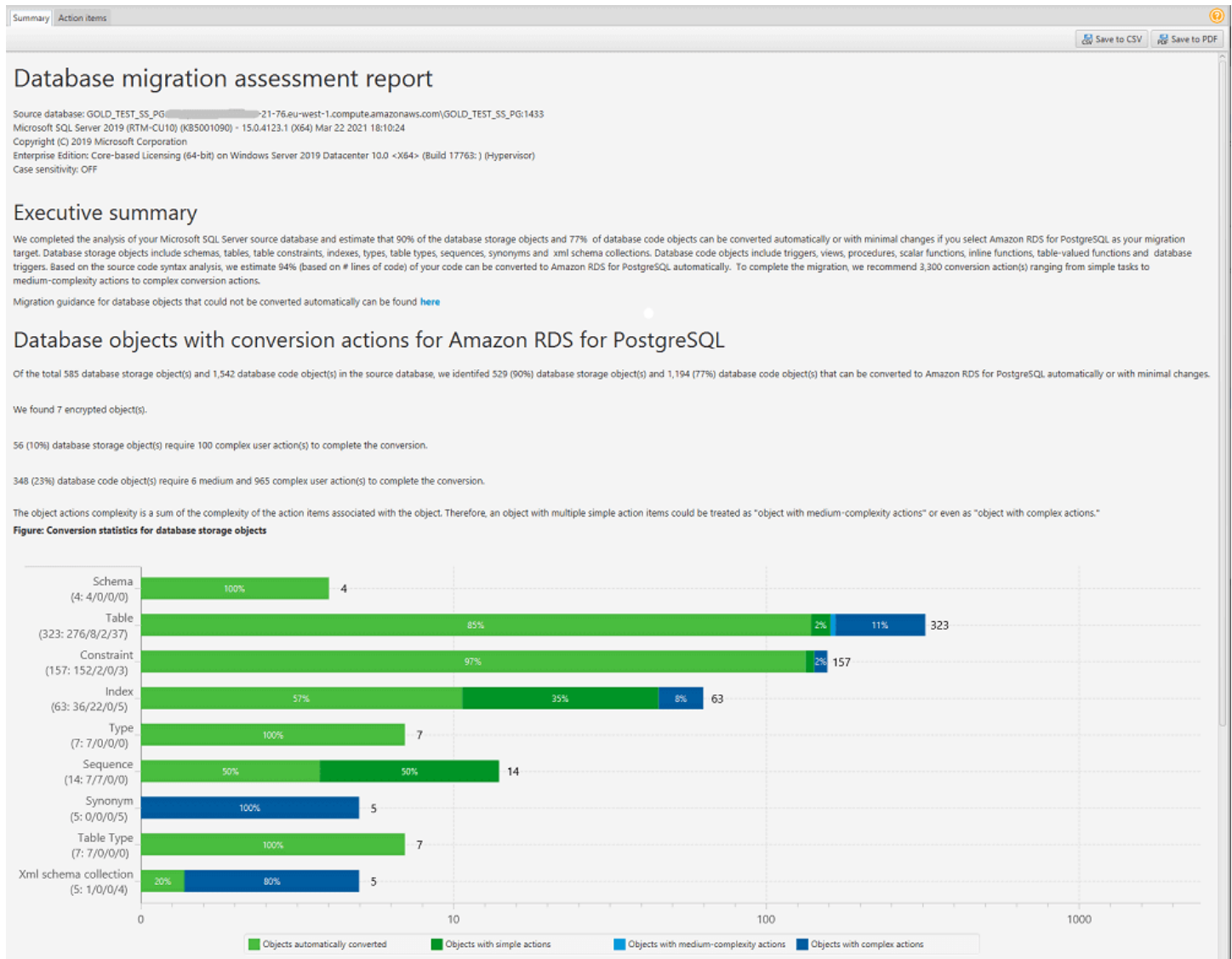
Creazione di un report di valutazione della migrazione del database

1. Nel pannello a sinistra, in cui viene visualizzato lo schema del database di origine, scegli un oggetto dello schema per cui creare un report di valutazione.
2. Apri il menu contestuale (tasto destro del mouse) per l'oggetto, quindi seleziona Create Report (Crea report).



Riepilogo del report di valutazione

Dopo aver creato un report di valutazione, si apre la visualizzazione del report di valutazione, contenente la scheda Summary (Riepilogo). Nella scheda Summary (Riepilogo) vengono visualizzate le informazioni di riepilogo del report di valutazione della migrazione del database. Essa mostra gli elementi convertiti automaticamente e quelli non convertiti automaticamente.



Per gli elementi dello schema che non possono essere convertiti automaticamente al motore di database di destinazione, il riepilogo include una stima dello sforzo necessario per creare elementi dello schema nell'istanza database di destinazione che siano equivalenti a quelli presenti nell'istanza database di origine.

Nel report, il tempo stimato per convertire questi elementi dello schema viene suddiviso nelle categorie seguenti:

- **Semplice:** azioni che possono essere completate in meno di un'ora.
- **Medio:** azioni più complesse che possono essere completate in una o quattro ore.
- **Significativa:** azioni molto complesse e che richiedono più di quattro ore per essere completate.

Elementi d'azione del rapporto di valutazione

La visualizzazione del report di valutazione include anche la scheda Action Items (Attività). Questa scheda contiene un elenco di elementi che non possono essere convertiti automaticamente nel motore di database del database di destinazione. Se selezioni un'attività dall'elenco, AWS SCT evidenzia l'elemento dallo schema al quale si applica l'attività.

Il report contiene anche le raccomandazioni per la conversione manuale dell'elemento dello schema. Per ulteriori informazioni circa la gestione delle conversioni manuali, consulta [Gestione delle conversioni manuali in AWS SCT](#).

The screenshot displays the AWS Schema Conversion Tool interface. The top navigation bar includes options like File, Actions, Assessment Report view, Settings, Applications, Help, Add source, and Add target. The main window is divided into several sections:

- Summary / Action items:** A tabbed interface with 'Action items' selected.
- Left Sidebar:** A tree view showing the source database structure:
 - Servers: SQL Server - ec2-21-76.eu-west-1.compute.
 - Databases [12]: AdventureWorks2012_CS, alfresco, GOLD_TEST_SS_PG, LARGE_DB_SS, master, model, msdb, tempdb.
 - TEST
 - Schemas [1]: dbo
 - Tables [8]: MSSQL_TemporalHistoryFor_1013578, MSSQL_TemporalHistoryFor_9655784, NonPartitionTable, PartitionTable, Position, test, tmp_tbl_sys_ver (System-Versioned), tmp_tbl_sys_ver_alter (System-Version).
 - Graph Tables
 - External Tables
 - Views
 - Procedures [4]: P_INS_OUTP, POSITION_UPDATE_CASH_CGT_BULK, test2, test3.
 - SQL scalar functions
 - SQL table-valued functions
 - SQL inline functions
- Main Content Area:** A list of issues with filters for Group by (Issues), Complexity (All), and Object type (All).
 - 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.
- Bottom Panel:** A detailed view of the selected issue (Issue 609) showing the source SQL code for a procedure:


```

1 create procedure POSITION_UPDATE_CASH_CGT_BULK
2   @InputPosNo tinyint unsigned readonly
3   , @posFlags bigint - 0
4   , @posFlagsMask bigint - 0
5 AS
6 update p
7 set   p.Flags = p.Flags & (~ @posFlagsMask ) | @posFlags
8 from   Position p
9       inner join @InputPosNo ipn on p.PosNo = ipn.F_POSNO
10
11 return 0
      
```
- Right Panel:** Target Amazon RDS for MySQL category: Schemas. Properties: Name, Value.

Salvataggio del rapporto di valutazione

Puoi salvare una copia locale del report di valutazione della migrazione del database sia come file PDF o come file con valori separati da una virgola (CSV). Il file CSV contiene solo le informazioni sulle attività. Il file PDF contiene sia i dati di riepilogo che le informazioni sull'azione, come illustrato nel seguente esempio.

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.

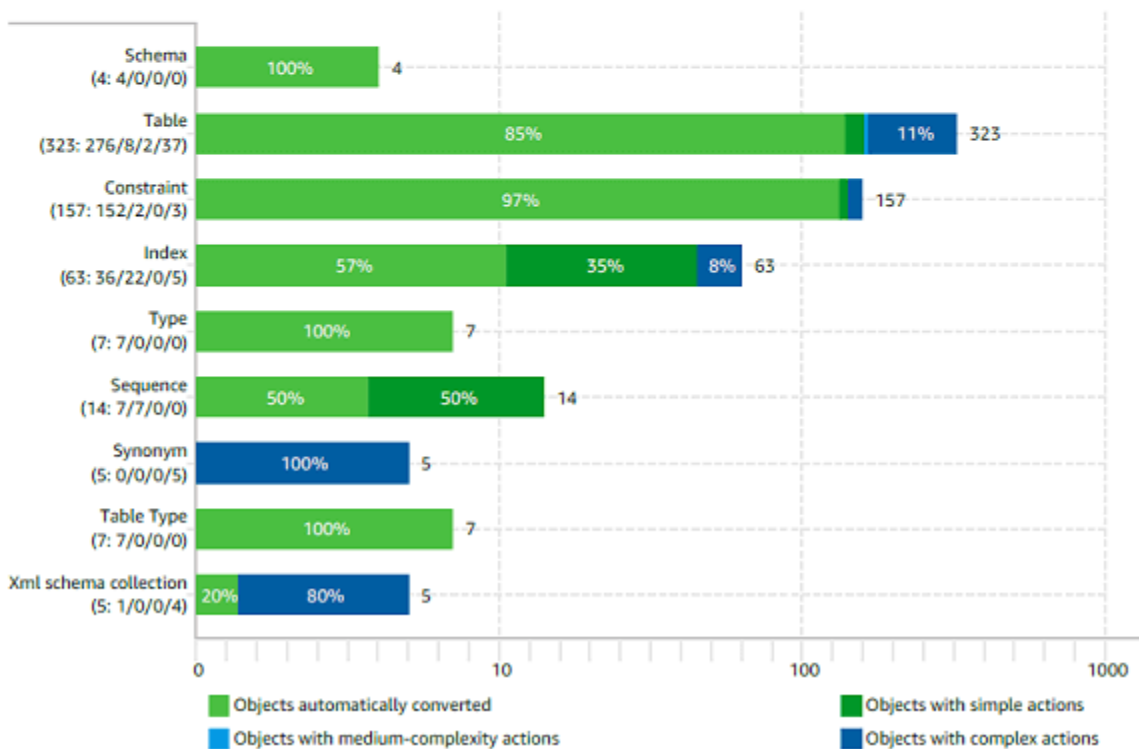
We found 7 encrypted object(s).

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



Gestione delle conversioni manuali in AWS SCT

Questo report di valutazione include un elenco di elementi che non possono essere convertiti automaticamente al motore di database del database di destinazione. Per ogni elemento che non può essere convertito, vi è un'azione nella scheda Action Items (Attività).

Puoi rispondere alle attività nel report di valutazione nei seguenti modi:

- Modifica dello schema del database di origine.

- Modifica dello schema del database di destinazione.

Modifica dello schema di origine

Per alcuni elementi, potrebbe essere più semplice modificare lo schema del database nel database di origine in uno schema che può essere convertito automaticamente. In primo luogo, verifica che le nuove modifiche siano compatibili con l'architettura della tua applicazione, quindi aggiorna lo schema del database di origine. Infine, aggiorna il progetto con le informazioni sullo schema aggiornato. Ora puoi convertire lo schema aggiornato e generare un nuovo report di valutazione della migrazione del database. Le attività non saranno più visibili per gli elementi modificati nello schema di origine.

Il vantaggio di questo processo è che lo schema aggiornato è sempre disponibile quando esegui l'aggiornamento dal database di origine.

Modifica dello schema di destinazione

Per alcuni elementi, potrebbe essere più semplice applicare lo schema convertito al database di destinazione, e in seguito aggiungere manualmente gli elementi dello schema equivalente al database di destinazione per gli elementi che non sono stati convertiti automaticamente. Puoi scrivere tutti gli schemi che possono essere convertiti automaticamente nel database di destinazione applicando lo schema. Per ulteriori informazioni, consulta [Salvare e applicare lo schema convertito in AWS SCT](#).

Gli schemi scritti nel database di destinazione non contengono gli elementi che non possono essere convertiti automaticamente. Dopo aver applicato lo schema al database di destinazione, in quest'ultimo puoi creare manualmente degli schemi equivalenti a quelli del database di origine. Le attività del report di valutazione della migrazione del database contengono suggerimenti per la creazione dello schema equivalente.

Warning

Se crei manualmente lo schema nel database di destinazione, salva una copia di qualsiasi operazione manuale venga realizzata. Se applichi di nuovo lo schema convertito del tuo progetto al database di destinazione, il lavoro manuale verrà sovrascritto.

In alcuni casi potresti non essere in grado di creare uno schema equivalente nel database di destinazione. Potrebbe essere necessario riprogettare una parte dell'applicazione e del database

per utilizzare la funzionalità disponibile nel motore del database di destinazione. In altri casi, puoi semplicemente ignorare lo schema che non può essere convertito automaticamente.

Aggiornamento e aggiornamento dello schema convertito in AWS SCT

Nel progetto AWS Schema Conversion Tool puoi aggiornare sia lo schema di origine che lo schema di destinazione.

- **Origine:** se si aggiorna lo schema per il database di origine, AWS SCT sostituisce lo schema del progetto con lo schema più recente del database di origine. Mediante questa funzionalità puoi aggiornare il progetto nel caso siano state apportate delle modifiche allo schema del database di origine.
- **Destinazione:** se si aggiorna lo schema per il database di destinazione, AWS SCT sostituisce lo schema del progetto con lo schema più recente del database di destinazione. Se non hai applicato alcuno schema al database di destinazione, AWS SCT elimina lo schema convertito dal progetto. Puoi quindi convertire lo schema del database di origine in un database di destinazione pulito.

Puoi aggiornare lo schema nel tuo AWS SCT progetto scegliendo **Aggiorna dal database**.

Salvare e applicare lo schema convertito in AWS SCT

Quando AWS Schema Conversion Tool genera lo schema convertito (come mostrato in [Convertire lo schema utilizzando AWS SCT](#)), lo schema convertito non viene applicato immediatamente all'istanza database di destinazione. Gli schemi convertiti rimangono invece archiviati localmente nel progetto fino a quando non sarai tu ad applicarli al database di destinazione. Mediante questa funzionalità puoi lavorare con gli elementi dello schema che non possono essere convertiti automaticamente nel motore di database di destinazione. Per ulteriori informazioni sugli elementi che non possono essere convertiti automaticamente, consulta [Creazione di report di valutazione della migrazione con AWS SCT](#).

Se lo desideri, puoi fare in modo che lo strumento salvi lo schema convertito in un file come script SQL prima di applicare lo schema al database di destinazione. Puoi anche fare in modo che lo strumento applichi lo schema convertito direttamente al database di destinazione.

Salvare lo schema convertito in un file

Puoi salvare lo schema convertito come script SQL in un file di testo. Utilizzando questo approccio, puoi modificare lo script SQL generato da AWS SCT per gestire gli elementi che lo strumento non è in grado di convertire automaticamente. Puoi quindi eseguire gli script aggiornati nell'istanza database di destinazione per applicare lo schema convertito al database di destinazione.

Per salvare lo schema convertito come script SQL

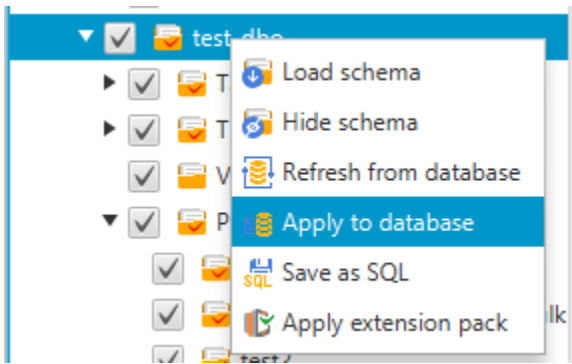
1. Scegli il tuo schema e apri il menu contestuale (clic con il pulsante destro del mouse).
2. Scegli Salva come SQL.
3. Inserisci il nome del file e scegli Salva.
4. Salva lo schema convertito utilizzando una delle seguenti opzioni:
 - File singolo
 - File singolo per fase
 - File singolo per dichiarazione

Per scegliere il formato dello script SQL

1. Nel menu Impostazioni, scegli Impostazioni progetto.
2. Scegli Salva script.
3. Per Vendor, scegli la piattaforma del database.
4. In Salva script SQL in, scegli come vuoi salvare lo script dello schema del database.
5. Scegliete OK per salvare le impostazioni.

Applicazione dello schema convertito

Quando sei pronto per applicare lo schema convertito al database destinazione, scegli l'elemento dello schema nel pannello a destra del progetto. Apri il menu contestuale (tasto destro del mouse) per l'elemento dello schema, quindi scegli Apply to database(Applica al database), come illustrato di seguito.



Lo schema del pacchetto di estensione

Quando applichi per la prima volta lo schema convertito all'istanza database di destinazione, AWS SCT aggiunge un ulteriore schema all'istanza database di destinazione. Questo schema implementa le funzioni di sistema del database di origine richieste durante la scrittura dello schema convertito nell'istanza database di destinazione. Lo schema viene chiamato schema del pacchetto di estensione.

Non modificare lo schema del pacchetto di estensione, in quanto questo potrebbe comportare risultati inattesi nello schema convertito scritto nell'istanza database di destinazione. Quando lo schema è completamente migrato all'istanza database di destinazione, e AWS SCT non è più necessario, puoi eliminare lo schema del pacchetto di estensione.

Lo schema del pacchetto di estensione viene denominato in base al database di origine come segue:

- Greenplum: `aws_greenplum_ext`
- Microsoft SQL Server: `aws_sqlserver_ext`
- Netezza: `aws_netezza_ext`
- Oracle: `aws_oracle_ext`
- Fiocco di neve: `aws_snowflake_ext`
- Teradata: `aws_teradata_ext`
- Vertica: `aws_vertica_ext`

Per ulteriori informazioni, consulta [Utilizzo dei pacchetti di AWS SCT estensione](#).

Librerie Python

Per creare funzioni personalizzate in Amazon Redshift, usi il linguaggio Python. Usa il pacchetto di AWS SCT estensione per installare le librerie python per il tuo database Amazon Redshift. Per ulteriori informazioni, consulta [Utilizzo dei pacchetti di AWS SCT estensione](#).

Ottimizzazione di Amazon Redshift utilizzando AWS SCT

Puoi utilizzarlo AWS Schema Conversion Tool per ottimizzare il tuo database Amazon Redshift. Utilizzando il database Amazon Redshift come origine e un database Amazon Redshift di prova come destinazione, AWS SCT consiglia chiavi di ordinamento e chiavi di distribuzione per ottimizzare il database.

Ottimizzazione del database Amazon Redshift

Utilizza la seguente procedura per ottimizzare il tuo database Amazon Redshift.

Per ottimizzare il tuo database Amazon Redshift

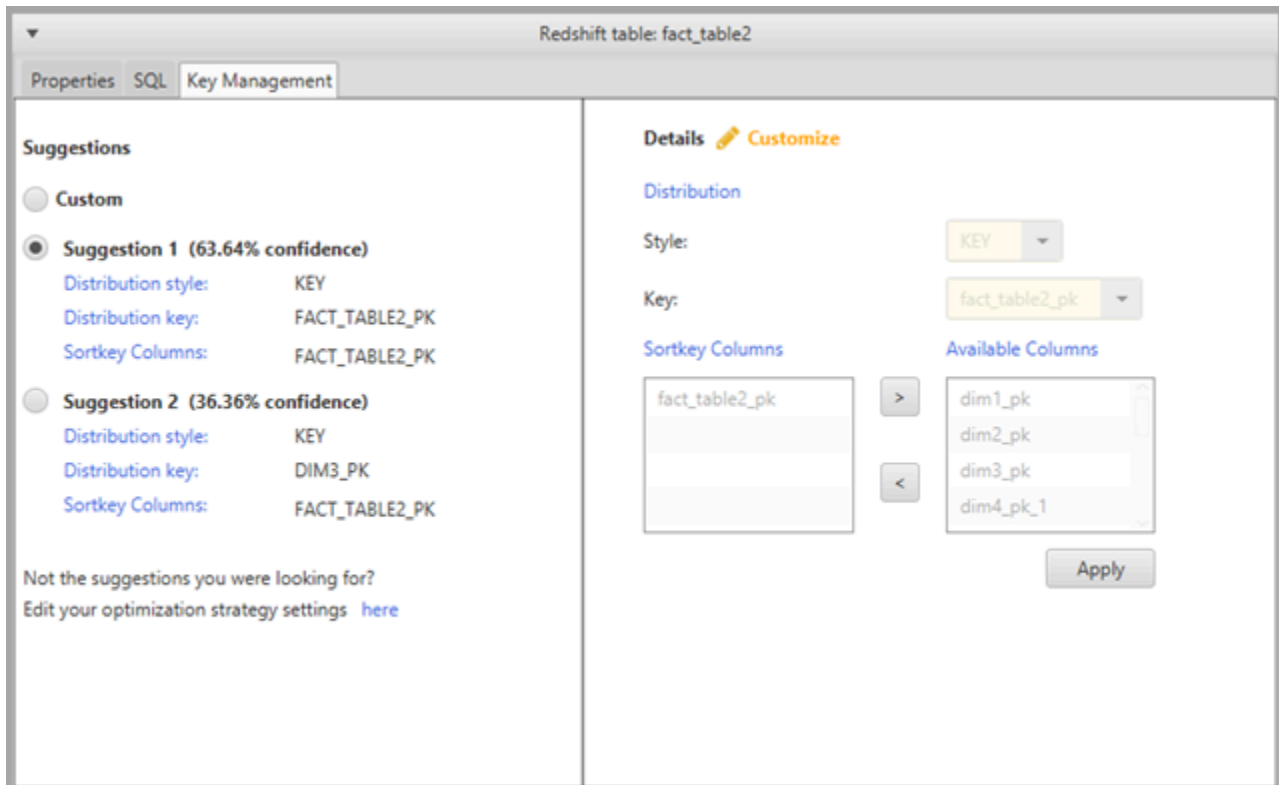
1. Scatta un'istantanea manuale del tuo cluster Amazon Redshift come backup. Puoi eliminare lo snapshot dopo aver completato l'ottimizzazione del tuo cluster Amazon Redshift e aver testato le modifiche apportate. Per ulteriori informazioni, consulta [Amazon Redshift snapshots](#).
2. Scegliere un oggetto dello schema da convertire nel pannello a sinistra del progetto. Aprire il menu contestuale (tasto destro del mouse) per l'oggetto, quindi scegliere Collect Statistics (Raccogli statistiche).

AWS SCT utilizza le statistiche per suggerire le chiavi di ordinamento e le chiavi di distribuzione.

3. Scegliere un oggetto dello schema da ottimizzare nel pannello a sinistra del progetto. Aprire il menu contestuale (tasto destro del mouse) per l'oggetto, quindi scegliere Run Optimization (Esegui ottimizzazione) .

AWS SCT suggerisce le chiavi di ordinamento e le chiavi di distribuzione.

4. Per verificare i suggerimenti, espandi il nodo delle tabelle sotto lo schema nel pannello a sinistra del progetto, quindi scegli una tabella. Scegliere la scheda Key Management (Gestione delle chiavi) come illustrato di seguito.



Il riquadro a sinistra contiene i suggerimenti per le chiavi e include il grado di affidabilità per ogni suggerimento. Puoi scegliere uno dei suggerimenti o personalizzare la chiave modificandola nel riquadro a destra.

5. Puoi creare un report contenente i suggerimenti di ottimizzazione. Per creare il report, procedi come di seguito:
 - a. Scegli un oggetto dello schema già ottimizzato nel pannello a sinistra del progetto. Apri il menu contestuale (tasto destro del mouse) per l'oggetto, quindi seleziona Create Report (Crea report).
 - Il report si apre nella finestra principale e viene visualizzata la scheda Summary (Riepilogo). Viene visualizzato nel report il numero degli oggetti con i suggerimenti di ottimizzazione.
 - b. Scegliere la scheda Action Items (Attività) per visualizzare i suggerimenti per le chiavi in un formato di report.
 - c. Puoi salvare una copia locale del report di ottimizzazione sia come file PDF o come file con valori separati da una virgola (CSV). Il file CSV contiene solo le informazioni sulle attività. Il file PDF contiene sia il riepilogo, sia le informazioni sulle attività.

6. Per applicare al database le ottimizzazioni consigliate, scegli un oggetto nel pannello a destra del progetto. Aprire il menu contestuale (tasto destro del mouse) per l'oggetto, quindi scegliere Apply to database (Applica al database).

Conversione dei processi di estrazione, trasformazione e caricamento (ETL) con AWS Schema Conversion Tool

Puoi usare il AWS Schema Conversion Tool (AWS SCT) per migrare i processi di estrazione, trasformazione e caricamento (ETL). Questo tipo di migrazione include la conversione della logica aziendale relativa a ETL. Questa logica può risiedere all'interno dei data warehouse di origine o in script esterni eseguiti separatamente.

Attualmente, AWS SCT supporta la conversione di script ETL in oggetti in AWS Glue e Amazon Redshift RSQL, come illustrato nella tabella seguente.

Origine	Target
Script ETL di Informatica	Informatica
Pacchetti ETL di Microsoft SQL Server Integrati on Services (SSIS)	AWS Glue o AWS Glue Studio
Script di shell con comandi incorporati da Teradata Basic Teradata Query (BTEQ)	RSQL Amazon Redshift
Script ETL Teradata BTEQ	AWS Glue o Amazon Redshift RSQL
Teradata FastExport script di lavoro	RSQL Amazon Redshift
Teradata FastLoad script di lavoro	RSQL Amazon Redshift
Teradata MultiLoad script di lavoro	RSQL Amazon Redshift

Argomenti

- [Conversione dei processi ETL in AWS Glue con AWS SCT](#)
- [Conversione dei processi ETL utilizzando l'API Python per AWS Glue con AWS SCT](#)
- [Conversione di script ETL di Informatica con AWS SCT](#)
- [Conversione di SSIS in AWS Glue con AWS SCT](#)
- [Conversione di SSIS in AWS Glue Studio con AWS SCT](#)

- [Conversione degli script Teradata BTEQ in Amazon Redshift RSQL conAWS SCT](#)
- [Conversione di script di shell con comandi Teradata BTEQ incorporati in Amazon Redshift RSQL conAWS SCT](#)
- [Conversione di TeradataFastExportscript di lavoro su Amazon Redshift RSQL conAWS SCT](#)
- [Conversione di TeradataFastLoadscript di lavoro su Amazon Redshift RSQL conAWS SCT](#)
- [Conversione di TeradataMultiLoadscript di lavoro su Amazon Redshift RSQL conAWS SCT](#)

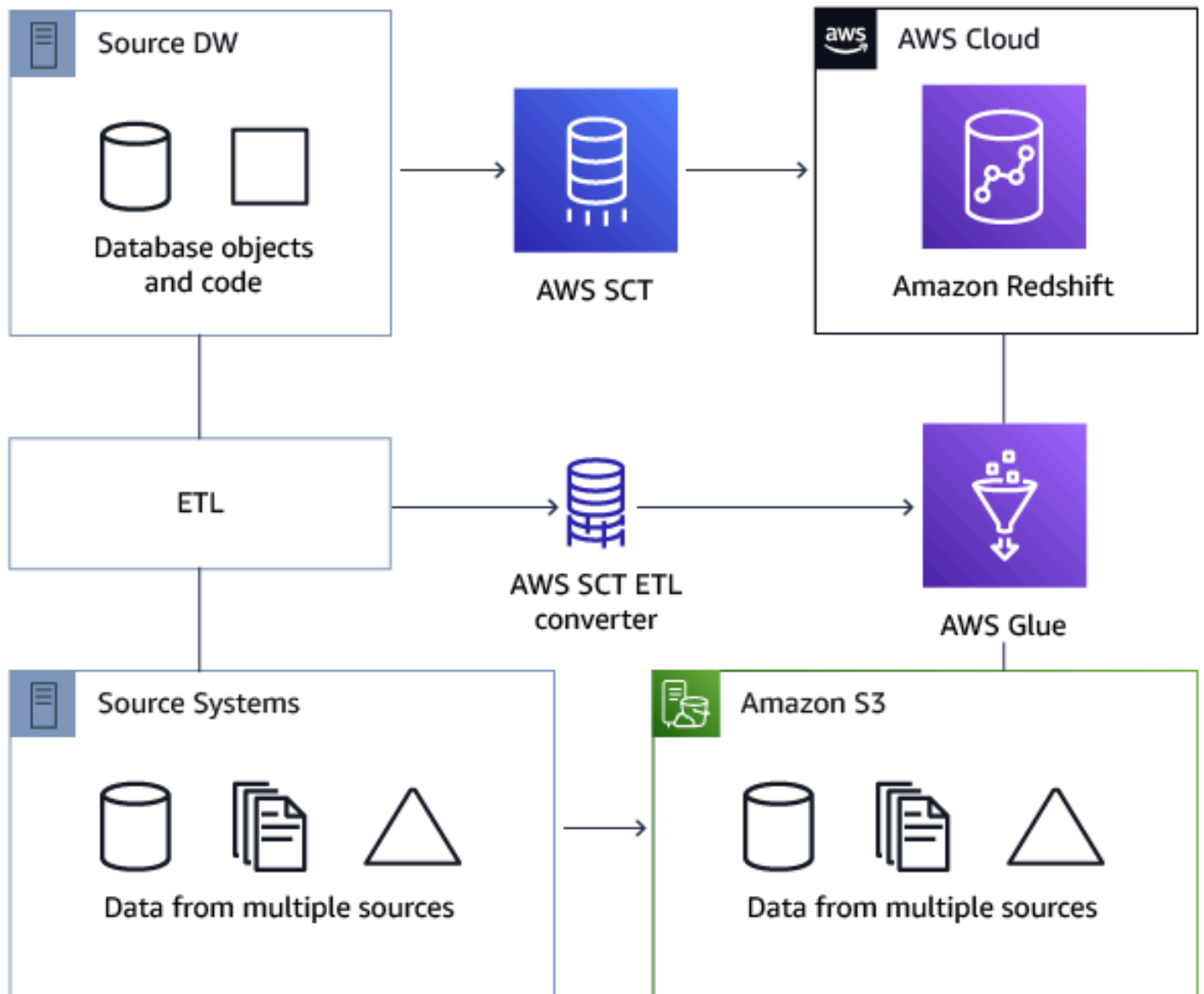
Conversione dei processi ETL inAWS GlueconAWS SCT

Di seguito, puoi trovare una descrizione del processo per convertire gli script ETL inAWS GlueconAWS SCT. Per questo esempio, convertiamo un database Oracle in Amazon Redshift, insieme ai processi ETL utilizzati con i database di origine e i data warehouse.

Argomenti

- [Prerequisiti](#)
- [Comprendere il catalogo dati di AWS Glue](#)
- [Limitazioni per la conversione utilizzandoAWS SCTconAWS Glue](#)
- [Fase 1: creazione di un nuovo progetto](#)
- [Fase 2: Creare unAWS Glueoccupazione](#)

Il seguente diagramma di architettura mostra un esempio di progetto di migrazione del database che include la conversione di script ETL inAWS Glue.



Prerequisiti

Prima di iniziare, esegui queste attività:

- Esegui la migrazione di tutti i database di origine a cui intendi migrare a AWS.
- Esegui la migrazione dei data warehouse di destinazione a AWS.
- Raccogliere un elenco di tutti i codici relativi al processo ETL.
- Raccogliere un elenco di tutte le informazioni necessarie relative alla connessione di ogni database.

Inoltre, AWS Glue necessita di autorizzazioni per accedere ad altri AWS risorse per tuo conto. È possibile fornire queste autorizzazioni usando AWS Identity and Access Management (IAM). Assicurati di aver creato una policy IAM per AWS Glue. Per ulteriori informazioni, vedere [Crea una policy IAM per AWS Servizio colla](#) nel AWS Glue Guida per gli sviluppatori.

Comprendere il catalogo dati di AWS Glue

Durante il processo di conversione, AWS Glue carica informazioni sui database di origine e di destinazione. Organizza queste informazioni in categorie, in una struttura chiamata albero. che include quanto segue:

- Connessioni— Parametri di connessione
- Crawler— Un elenco di crawler, un crawler per ogni schema
- Banche dati— Contenitori che contengono tavoli
- Tavoli— Definizioni di metadati che rappresentano i dati nelle tabelle
- Offerte di lavoro ETL— Logica aziendale che esegue il lavoro ETL
- Trigger— Logica che controlla quando viene eseguito un processo ETL AWS Glue (su richiesta, in base alla pianificazione o in base a eventi lavorativi)

Il catalogo dati di AWS Glue è un indice per la posizione, lo schema e i parametri di runtime dei dati. Quando utilizzi AWS Glue e AWS SCT, il catalogo dati di AWS Glue contiene riferimenti ai dati utilizzati come origini e destinazioni dei processi ETL in AWS Glue. Per creare il tuo data warehouse, cataloga questi dati.

È possibile utilizzare le informazioni presenti nel catalogo dati per creare e monitorare i processi ETL. In genere, per richiedere l'inventario dei dati contenuti nel data store, si esegue un crawler. Tuttavia, è possibile aggiungere tabelle di metadati nel catalogo dati in altri modi.

Quando una tabella viene definita nel catalogo dati, viene aggiunta a un database. Un database viene usato per organizzare le tabelle in AWS Glue.

Limitazioni per la conversione utilizzando AWS SCT con AWS Glue

Le conversioni eseguite tramite AWS SCT con AWS Glue sono soggette alle limitazioni indicate di seguito.

Resource (Risorsa)	Limite predefinito
--------------------	--------------------

Numero di database per ogni account	10.000
Numero di tabelle per ogni database	100.000
Numero di partizioni per ogni tabella	1.000.000
Numero di versioni di tabella per ogni tabella	100.000
Numero di tabelle per ogni account	1.000.000
Numero di partizioni per ogni account	10.000.000
Numero di versioni di tabella per ogni account	1.000.000
Numero di connessioni per ogni account	1.000
Numero di crawler per ogni account	25
Numero di processi per ogni account	25
Numero di trigger per ogni account	25
Numero di esecuzioni di processo simultanee per ogni account	30
Numero di esecuzioni di processo simultanee per ogni processo	3
Numero di processi per ogni trigger	10
Numero di endpoint di sviluppo per ogni account	5
Numero massimo di unità di elaborazione dati (DPU) utilizzate contemporaneamente da un endpoint di sviluppo	5
Numero massimo di DPU utilizzate simultaneamente da un ruolo	100

Lunghezza del nome del database	Illimitato Per compatibilità con altri store di metadati, ad esempio Apache Hive, il nome viene trasformato perché sia in caratteri minuscoli. Se prevedi di accedere al database da Amazon Athena, fornisci un nome con solo caratteri alfanumerici e caratteri di sottolineatura.
Lunghezza del nome della connessione	Illimitato
Lunghezza del nome del crawler	Illimitato

Fase 1: creazione di un nuovo progetto

Per creare un nuovo progetto, segui questi passaggi di alto livello:

1. Crea un nuovo progetto in AWS SCT. Per ulteriori informazioni, consulta [Creare unAWS SCT progetto](#).
2. Aggiungi i tuoi database di origine e di destinazione al progetto. Per ulteriori informazioni, consulta [Aggiungere server di database a unAWS SCT progetto](#).

Assicurati di aver scelto l'uso di AWS Glue nelle impostazioni di connessione al database di destinazione. Per farlo, scegli l'AWS Glue lingua. Per la copia da AWS, scegli il profilo che desideri utilizzare. Il profilo dovrebbe compilare automaticamente la chiave di accesso AWS, chiave segreta e cartella bucket Amazon S3. In caso contrario, inserisci queste informazioni. Dopo aver scelto OK, AWS Glue analizza gli oggetti e carica i metadati nel catalogo dati di AWS Glue.

A seconda delle impostazioni di sicurezza, potrebbe essere visualizzato un messaggio di errore che segnala che il tuo account non dispone dei privilegi sufficienti per alcuni schemi sul server. Se disponi dell'accesso agli schemi in uso, puoi ignorare il messaggio.

3. Per completare la preparazione all'importazione del tuo ETL, connettiti ai database di origine e di destinazione. A tale scopo, scegli il tuo database nell'albero dei metadati di origine o di destinazione, quindi scegli di connetterti al server.

AWS Glue crea un database sul server del database di origine e uno sul server del database di destinazione per facilitare la conversione ETL. Il database nel server di destinazione contiene il catalogo dati di AWS Glue. Per trovare oggetti specifici, utilizzate la ricerca nei pannelli di origine o di destinazione.

Per vedere come viene convertito un oggetto specifico, trova l'elemento che desideri convertire e scegli **Converti schema** dal relativo menu contestuale (tasto destro). AWS SCT trasforma questo oggetto selezionato in uno script.

È possibile rivedere lo script convertito dal **Script** cartella nel pannello di destra. Attualmente, lo script è un oggetto virtuale, disponibile solo come parte del tuo AWS SCT progetto.

Per creare un AWS Glue lavoro con il tuo script convertito, carica il tuo script su Amazon S3. Per caricare lo script su Amazon S3, scegli lo script, quindi scegli **Salva su S3** dal relativo menu contestuale (tasto destro).

Fase 2: Creare un AWS Glue occupazione

Dopo aver salvato lo script su Amazon S3, puoi sceglierlo e quindi scegliere **Configurare AWS Glue Lavoro** per aprire la procedura guidata per configurare AWS Glue lavoro. La procedura guidata semplifica la configurazione:

1. Nella prima scheda del wizard, **Flusso di dati di progettazione**, puoi scegliere una strategia di esecuzione e l'elenco degli script che desideri includere in questo unico lavoro. È possibile scegliere i parametri per ogni script. È anche possibile ridisporre gli script in modo che vengano eseguiti nell'ordine corretto.
2. Nella seconda scheda, puoi assegnare un nome al tuo lavoro e configurare direttamente le impostazioni per AWS Glue. Su questa schermata, è possibile configurare le seguenti impostazioni:
 - AWS Identity and Access Management ruolo (IAM)
 - Nomi file script e percorsi di file
 - Crittografia lo script utilizzando la crittografia lato server con chiavi gestite da Amazon S3 (SSE-S3)
 - Directory temporanea
 - Percorso libreria Python generato
 - Percorso utente libreria Python
 - Percorso per i file.jar dipendenti
 - Percorso dei file di riferimento

- DPU simultanee per ogni processo eseguito
- Simultaneità massima
- Timeout del processo (in minuti)
- Soglia notifica ritardo (in minuti)
- Numero di tentativi
- Configurazione di sicurezza
- Crittografia lato server

3. Nella terza fase o scheda, scegli la connessione configurata verso l'endpoint di destinazione.

Dopo averne completato la configurazione, il processo verrà visualizzato insieme ai processi ETL nel catalogo dati di AWS Glue. Se scegli il processo, vengono visualizzate le relative impostazioni in modo da controllarle o modificarle. Per creare un nuovo lavoro in AWS Glue, scegli **Crea AWS Glue Lavoro** dal menu contestuale (clic con il pulsante destro del mouse) del lavoro. In questo modo viene applicata la definizione dello schema. Per aggiornare la visualizzazione, scegli **Refresh from database (Aggiorna dal database)** dal menu contestuale (tasto destro del mouse).

A questo punto puoi visualizzare il processo nella console AWS Glue. Per farlo, accedi all'AWS Management Console e apri la console AWS Glue a <https://console.aws.amazon.com/glue/>.

Puoi testare il nuovo processo per verificare che funzioni correttamente. A questo scopo, controlla i dati nella tabella di origine, quindi verifica che la tabella di destinazione sia vuota. Esegui il processo e verifica di nuovo. Puoi visualizzare i log degli errori nella console di AWS Glue.

Conversione dei processi ETL utilizzando l'API Python per AWS Glue con AWS SCT

Nelle seguenti sezioni viene descritta una conversione che chiama le operazioni API di AWS Glue in Python. Per ulteriori informazioni, vedere [Programma AWS Glue Script ETL in Python](#) nel AWS Glue Guida per gli sviluppatori.

Argomenti

- [Fase 1: crea un database](#)
- [Fase 2: Creare una connessione](#)
- [Fase 3: Creare un AWS Glue catalogato](#)

Fase 1: crea un database

Il primo passo consiste nel creare un nuovo database in unAWS GlueCatalogo dati utilizzando il[AWSAPI SDK](#). Quando una tabella viene definita nel catalogo dati, viene aggiunta a un database. Per organizzare le tabelle in AWS Glue viene utilizzato un database.

L'esempio seguente illustra il metodo `create_database` dell'API Python per AWS Glue.

```
response = client.create_database(  
    DatabaseInput={  
        'Name': 'database_name',  
        'Description': 'description',  
        'LocationUri': 'string',  
        'Parameters': {  
            'parameter-name': 'parameter value'  
        }  
    }  
)
```

Se utilizzi Amazon Redshift, il nome del database è formato come segue.

```
{redshift_cluster_name}_{redshift_database_name}_{redshift_schema_name}
```

Il nome completo del cluster Amazon Redshift per questo esempio è il seguente.

```
rsdbb03.apq1mpqso.us-west-2.redshift.amazonaws.com
```

L'esempio seguente mostra un nome di database valido. In questo caso, `rsdbb03` è il nome, che è la prima parte del nome completo dell'endpoint del cluster. Il nome del database è `dev` e lo schema è `ora_glue`.

```
rsdbb03_dev_ora_glue
```

Fase 2: Creare una connessione

Creare una nuova connessione in un catalogo dati utilizzando il[AWSAPI SDK](#).

L'esempio seguente illustra il metodo `create_connection` dell'API Python per 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'
    ]
  }
}
)

```

I parametri utilizzati in `create_connection` sono i seguenti:

- **Name**(stringa UTF-8) — obbligatorio. Per Amazon Redshift, il nome della connessione è formato come segue: `Redshift_<Endpoint-name>_<redshift-database-name>`, ad esempio: `Redshift_abcde03_dev`
- **Description**(stringa UTF-8) — La tua descrizione della connessione.
- **ConnectionType**(Stringa UTF-8) — Obbligatoria. Il tipo di connessione. Al momento è supportata solo JDBC; SFTP non è supportata.
- **ConnectionProperties**(dict) — Obbligatorio. Un elenco di coppie chiave-valore utilizzate come parametri per questa connessione, incluso l'URL della connessione JDBC, il nome utente e la password.
- **PhysicalConnectionRequirements**(dict) — Requisiti di connessione fisica, che includono quanto segue:
 - **SubnetId**(stringa UTF-8) — L'ID della sottorete utilizzata dalla connessione.
 - **SecurityGroupIdList**(elenco) — L'elenco degli ID del gruppo di sicurezza utilizzato dalla connessione.
 - **AvailabilityZone**(Stringa UTF-8) — Obbligatoria. La zona di disponibilità che contiene l'endpoint. Questo parametro è obsoleto.

Fase 3: Creare unAWS Gluecatalogo

Ora procediamo con la creazione di un crawler AWS Glue per compilare il catalogo AWS Glue. Per ulteriori informazioni, vedere [Catalogazione delle tabelle con un crawler](#) nelAWS GlueGuida per gli sviluppatori.

Il primo passaggio per aggiungere un crawler consiste nel creare un nuovo database in un catalogo dati utilizzando il [AWSAPI SDK](#). Prima di iniziare, assicurati di eliminarne qualsiasi versione precedente utilizzando il `delete_crawler` operazione.

Durante la creazione di un crawler è necessario tenere conto di alcuni aspetti:

- Per il nome del crawler è necessario utilizzare il formato `<redshift_node_name>_<redshift_database_name>_<redshift_shema_name>`, ad esempio: `abcde03_dev_ora_glue`
- È necessario utilizzare un ruolo IAM già esistente. Per ulteriori informazioni sulla creazione di ruoli IAM, consulta [Creazione di ruoli IAM](#) nelGuida per l'utente IAM.
- È necessario utilizzare il nome del database creato nelle fasi precedenti.
- Il parametro `ConnectionName` va utilizzato perché è obbligatorio.
- Per il parametro `path`, occorre utilizzare il percorso della destinazione JDBC, ad esempio: `dev/ora_glue/%`

L'esempio seguente elimina un crawler esistente, quindi ne crea uno nuovo, utilizzando l'API Python per AWS 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'
)

```

Crea e quindi esegui un crawler che si collega a uno o più datastore, determina le strutture dei dati e scrive tabelle nel catalogo dati. È possibile eseguire il crawler secondo un programma, come illustrato di seguito.

```

response = client.start_crawler(
    Name='string'
)

```

Questo esempio utilizza Amazon Redshift come obiettivo. I tipi di dati di Amazon Redshift sono mappati aAWS Glue digitati i dati nel modo seguente dopo l'esecuzione del crawler.

Tipo di dati Amazon Redshift	Tipo di dati AWS Glue
smallint	smallint

integer	int
bigint	bigint
decimal	decimal(18,0)
decimal(p,s)	decimal(p,s)
real	double
double precision	double
Booleano	Booleano
char	Stringa
varchar	Stringa
varchar(n)	Stringa
date	date
timestamp	timestamp
timestampz	timestamp

Conversione di script ETL di Informatica con AWS SCT

Puoi usare AWS SCT interfaccia a riga di comando (CLI) per convertire gli script ETL di Informatica in modo da poterli utilizzare con il nuovo database di destinazione. Questa conversione include tre passaggi chiave. Primo, AWS SCT converte il codice SQL incorporato negli oggetti Informatica. Avanti, AWS SCT modifica i nomi degli oggetti del database in base alle regole di migrazione specificate nel progetto. Infine, AWS SCT reindirizza le connessioni degli script ETL di Informatica al nuovo database di destinazione.

È possibile convertire gli script ETL di Informatica come parte di AWS SCT progetto di conversione del database. Assicuratevi di aggiungere i database di origine e di destinazione al progetto quando convertite gli script ETL di Informatica.

Per convertire gli script ETL di Informatica, assicurati di utilizzare AWS SCT versione 1.0.667 o superiore. Inoltre, acquisisci familiarità con l'interfaccia a riga di comando di AWS SCT. Per ulteriori informazioni, consulta [AWS SCT Riferimento CLI](#).

Per convertire gli script ETL di Informatica utilizzando AWS SCT

1. Crea un nuovo AWS SCT Crea script CLI o modifica un modello di scenario esistente. Ad esempio, puoi scaricare e modificare `InformaticConversionTemplate.sct` modello. Per ulteriori informazioni, consulta [Ottenere scenari CLI](#).
2. Scarica i driver JDBC richiesti per i tuoi database di origine e di destinazione. Specificare la posizione di questi driver utilizzando `SetGlobalSettings` comando. Inoltre, specifica le cartelle in cui AWS SCT può salvare i file di registro.

L'esempio di codice seguente mostra come aggiungere il percorso dei driver Oracle e PostgreSQL al AWS SCT impostazioni. Dopo aver eseguito questo esempio di codice, AWS SCT memorizza i file di registro nel `C:\sct_log` cartella. Inoltre, AWS SCT memorizza i file di registro della console nel `C:\Temp\oracle_postgresql` cartella.

```
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. Crea un nuovo AWS SCT progetto. Inserisci il nome e la posizione del tuo progetto.

L'esempio di codice seguente crea `oracle_postgresql` progetto nel `C:\Temp` cartella.

```
CreateProject
  -name: 'oracle_postgresql'
  -directory: 'C:\Temp'
/
```

4. Aggiungi informazioni di connessione sui database di origine e di destinazione.

L'esempio di codice seguente aggiunge i database Oracle e PostgreSQL come origine e destinazione per ilAWS SCTprogetto.

```
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'
/
```

Nell'esempio precedente, sostituisci *utente_sorgente* e *utente_destinazione* con i nomi degli utenti del tuo database. Quindi, sostituisci *password di origine* e *password_destinazione* con le tue password.

Per *indirizzo_origine* e *indirizzo_destinazione*, inserisci gli indirizzi IP dei server del database di origine e di destinazione.

Per connettersi a un database Oracle versione 19 e successive, utilizzare il nome del servizio Oracle nel `AddSource` comando. A tale scopo, aggiungi il `-connectionType` parametro e impostane il valore su `'basic_service_name'`. Quindi, aggiungi il `-serviceName` parametro e impostane il valore sul nome del servizio Oracle. Per ulteriori informazioni su `AddSource` comando, vedi [AWS Schema Conversion Tool Riferimento CLI](#).

5. Crea un nuovoAWS SCTregola di mappatura, che definisce i motori del database di destinazione per ogni schema del database di origine. Per ulteriori informazioni, consulta [Creazione di regole di mappatura in AWS SCT](#).

L'esempio di codice seguente crea una regola di mappatura che include tutti gli schemi del database Oracle di origine e definisce PostgreSQL come obiettivo di migrazione.

```
AddServerMapping
  -sourceTreePath: 'Servers.ORACLE'
  -targetTreePath: 'Servers.POSTGRESQL'
/
```

6. Aggiungi informazioni di connessione sui file XML di origine e destinazione di Informatica.

L'esempio di codice seguente aggiunge i file XML di Informatica daC:

`\Informatica_sourceeC:\Informatica_targetcartelle.`

```
AddSource
  -name: 'INFA_SOURCE'
  -vendor: 'INFORMATICA'
  -mappingsFolder: 'C:\Informatica_source'
/
AddTarget
  -name: 'INFA_TARGET'
  -vendor: 'INFORMATICA'
  -mappingsFolder: 'C:\Informatica_target'
/
```

7. Crea un'altra regola di mappatura per definire il file XML Informatica di destinazione per il tuo file XML Informatica di origine.

L'esempio di codice seguente crea una regola di mappatura che include i file XML Informatica di origine e di destinazione utilizzati nell'esempio precedente.

```
AddServerMapping
  -sourceTreePath: 'ETL.INFA_SOURCE'
  -targetTreePath: 'ETL.INFA_TARGET'
/
```

8. Specificare la connessione al server del database che corrisponde al riferimento del nome della connessione Informatica.

Il seguente esempio di codice configura il reindirizzamento degli script ETL di Informatica dall'origine al nuovo database di destinazione. Questo esempio configura anche le variabili di connessione.

```
ConfigureInformaticaConnectionsRedirect
  -treePath: 'ETL.INFA_SOURCE.Files'
```

```
-connections: '{
"ConnectionNames": [
{
"name": "Oracle_src",
"newName": "postgres",
"treePath": "Servers.ORACLE"
}
]
"ConnectionVariables": [
{
"name": "$Source",
"treePath": "Servers.ORACLE"
}
]
}'
/
```

9. Converti gli schemi del database di origine e gli script ETL di Informatica.

Il seguente esempio di codice converte tutti gli schemi del database Oracle di origine e il file XML Informatica.

```
Convert
-treePath: 'Servers.ORACLE.Schemas.%'
/
Convert
-treePath: 'ETL.INFA_SOURCE.Files'
/
```

10. (Facoltativo) Salva il progetto di conversione e il rapporto di valutazione. Questo rapporto include le azioni di conversione e i consigli su come affrontarle.

L'esempio di codice seguente salva il progetto e salva una copia del rapporto di valutazione come file PDF nelC:\Tempcartella.

```
SaveProject
/
SaveReportPDF
-treePath: 'ETL.INFA_SOURCE.Files'
-file: 'C:\Temp\Informatica.pdf'
/
```

11. Salvate il file XML Informatica convertito.

L'esempio di codice seguente salva il file XML convertito nelC:\Tempcartella. Hai specificato questa cartella nel passaggio precedente utilizzando ilAddTargetcomando.

```
SaveTargetInformaticaXML
-treePath: 'ETL.INFA_TARGET.Files'
/
```

12. Salva il tuo script come .sctsfile ed esegilo usando ilRunSCTBatchcomando nelAWS SCTCLIP. Per ulteriori informazioni, consulta [AWS SCT modalità script CLI](#).

L'esempio seguente esegue ilInformatica.sctssceneggiatura nelC:\Tempcartella. Puoi usare questo esempio in Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\Temp\Informatica.scts"
```

Se modifichi gli script ETL di Informatica di origine, eseguiAWS SCTDi nuovo script CLI.

Conversione di SSIS inAWS GlueconAWS SCT

Di seguito, puoi scoprire come convertire i pacchetti Microsoft SQL Server Integration Services (SSIS) inAWS GlueutilizzandoAWS SCT.

Per convertire i pacchetti Microsoft SSIS inAWS Glue, assicurati di utilizzareAWS SCTversione 1.0.642 o superiore. È inoltre necessario disporre di un progetto SSIS con pacchetti ETL:.dtsx,.conmgr, e .paramsfile nella cartella locale.

Non è necessario un server SSIS installato. Il processo di conversione passa attraverso i file SSIS locali.

Per convertire un pacchetto SSIS inAWS GlueutilizzandoAWS SCT

1. Crea un nuovo progetto inAWS SCTo apri un progetto esistente. Per ulteriori informazioni, consulta [the section called "Creazione di un progetto"](#).
2. ScegliAggiungi fontedal menu per aggiungere un nuovo pacchetto SSIS sorgente al tuo progetto.
3. ScegliServizi di integrazione SQL Servere completa quanto segue:
 - Nome della connessione— Inserisci il nome della tua connessione.AWS SCTvisualizza questo nome nell'albero dei metadati.

- cartella dei pacchetti SSIS— Scegli il percorso della cartella del tuo progetto SSIS con i pacchetti.

AWS SCT legge i file del progetto (file con le estensioni) .dtsx, .conmgro.params) dalla cartella locale e li analizza. Quindi li organizza in unAWS SCTalbero delle categorie.

4. ScegliAggiungi obiettivi dal menu per aggiungere una nuova piattaforma di destinazione per convertire i pacchetti SSIS di origine.
5. ScegliAWS Gluee completa quanto segue:
 - Nome della connessione— Inserisci il nome della tua connessione.AWS SCTvisualizza questo nome nell'albero dei metadati.
 - Copia daAWSprofilo— Scegli il profilo da utilizzare.
 - AWSchiave di accesso— Inserisci il tuoAWSchiave di accesso.
 - AWSchiave segreta— Inserisci il tuoAWSchiave segreta.
 - Regione— Scegli ilRegione AWSche desideri utilizzare dall'elenco.
 - Cartella bucket Amazon S3— Inserisci il percorso della cartella per il bucket Amazon S3 che intendi utilizzare.

Puoi usare un virtualeAWS Gluebersaglio. In questo caso, non è necessario specificare le credenziali di connessione. Per ulteriori informazioni, consulta [the section called “Obiettivi virtuali”](#).

6. Crea una nuova regola di mappatura che includa il tuo pacchetto SSIS di origine e il tuoAWS Gluebersaglio. Per ulteriori informazioni, consulta [the section called “Nuova regola”](#).
7. SulVisualizzamenu, scegliVista principale.
8. Nella vista ad albero SSIS, apri il menu contestuale (clic con il pulsante destro del mouse) perGestori delle connessioni, quindi scegliConfigurazione delle connessioni.
9. Configura il gestore della connessione del progetto.

Per configurare una mappatura delle connessioni per i gestori delle connessioni SSIS, specificareAWS Glueconnessione per il gestore della connessione SSIS corrispondente. Assicurati che il tuoAWS Gluele connessioni sono già state create.

- a. SottoConnessioni, scegliConnessioni al progetto.
- b. PerCollegamento al catalogo Glue, scegli l'appropriatoAWS Glueconnessione.

10. Configura il gestore della connessione dei pacchetti:
 - a. SottoConnessioni, scegli il tuo pacchetto.
 - b. PerCollegamento al catalogo Glue, scegli l'appropriatoAWS Glueconnessione.
 - c. Ripeti queste azioni per tutte le connessioni disponibili per il tuo pacchetto.
11. Seleziona Apply (Applica).
12. Converti il tuo pacchetto. Nella vista ad albero dei sorgenti, trovaPacchetti. Apri il menu contestuale (clic con il pulsante destro del mouse) per il tuo pacchetto, quindi scegliConverti pacchetto.
13. Salva lo script convertito su Amazon S3. Nella vista ad albero del bersaglio, trovaScript di pacchetti. Apri il menu contestuale (clic con il pulsante destro del mouse) per lo script convertito, quindi scegliSalva su S3.
14. Configura il tuoAWS Gluelavoro. Nella vista ad albero del bersaglio, trovaScript di pacchetti. Apri il menu contestuale (clic con il pulsante destro del mouse) per lo script convertito, quindi scegliConfigurareAWS Glueoccupazione.
15. Completa le tre sezioni di configurazione:
 - a. Completa ilFlusso di dati di progettazioneesezione:
 - Strategia di esecuzione— Scegli in che modo il tuo lavoro eseguirà gli script ETL. ScegliSEQUENZIALEper eseguire gli script nell'ordine specificato nella procedura guidata. ScegliPARALLELOper eseguire gli script in parallelo, ignorando l'ordine specificato nella procedura guidata.
 - Script— Scegli il nome del tuo script convertito.
 - Seleziona Successivo.
 - b. Completa ilProprietà lavorativeesezione:
 - Nome— Inserisci il nome del tuoAWS Gluelavoro.
 - Ruolo IAM— Scegli il ruolo IAM utilizzato per l'autorizzazione alle risorse utilizzate per eseguire il lavoro e accedere agli archivi dati.
 - Nome del file di script— Inserisci il nome dello script convertito.
 - Percorso S3 del file di script— Inserisci il percorso Amazon S3 dello script convertito.
 - Crittografa lo script utilizzando SSE-S3— Scegli questa opzione per proteggere i dati utilizzando la crittografia lato server con chiavi di crittografia gestite da Amazon S3 (SSE-S3).

- Elenco temporaneo— Inserisci il percorso Amazon S3 in una directory temporanea per risultati intermedi. AWS Glue e AWS Glue ETL trasformazioni integrate utilizzano questa directory per leggere o scrivere su Amazon Redshift.
 - AWS SCT genera automaticamente il percorso per le librerie Python. Puoi rivedere questo percorso in Percorso della libreria python generato. Non puoi modificare questo percorso generato automaticamente. Per utilizzare librerie Python aggiuntive, inserisci il percorso in Percorso della libreria python dell'utente.
 - Percorso della libreria python dell'utente— Inserisci i percorsi per le librerie Python aggiuntive degli utenti. Separa i percorsi Amazon S3 con una virgola.
 - Percorso dipendente dei vasi— Inserisci i percorsi per i file jar dipendenti. Separa i percorsi Amazon S3 con una virgola.
 - Percorso dei file di riferimento— Inserisci i percorsi per i file aggiuntivi, come i file di configurazione, richiesti dallo script. Separa i percorsi Amazon S3 con una virgola.
 - Capacità massima— Inserisci il numero massimo di unità di elaborazione dati (DPU) che possono essere allocate durante l'esecuzione di questo processo. È possibile immettere un numero intero compreso tra 2 e 100. Il valore predefinito è 2.
 - Concorrenza massima— Inserire il numero massimo di esecuzioni simultanee consentite per questo processo. L'impostazione predefinita è 1. AWS Glue restituisce un errore quando viene raggiunta questa soglia.
 - Timeout del lavoro (minuti)— Inserisci il valore di timeout sul tuo processo ETL come protezione contro i lavori fuori produzione. Il valore predefinito è 2.880 minuti (48 ore) per i processi batch. Se il processo supera questo limite, lo stato di esecuzione del processo cambia in TIMEOUT.
 - Soglia di notifica di ritardo (minuti)— Inserisci la soglia pochi minuti prima AWS SCT invia una notifica di ritardo.
 - Numero di tentativi— Inserisci il numero di volte (0-10) che AWS Glue dovrebbe riavviare automaticamente il processo in caso di errore. I lavori che raggiungono il limite di timeout non vengono riavviati. Il valore predefinito è 0.
 - Seleziona Successivo.
- c. Configura le connessioni richieste:
- i. Da Tutte le connessioni, scegli il richiesto AWS Glue connessioni e aggiungile all'elenco di Connessioni selezionate.
 - ii. Scegli Finish (Fine).

16. Crea un file configurato AWS Glue lavoro. Nella vista ad albero di destinazione, trova ed espandi Offerte di lavoro ETL. Apri il menu contestuale (clic con il pulsante destro del mouse) per il job ETL che hai configurato, quindi scegli Crea AWS Glue Lavoro.
17. Esegui il AWS Glue lavoro:
 - a. Apri la console AWS Glue all'indirizzo <https://console.aws.amazon.com/glue/>.
 - b. Nel riquadro di navigazione scegliere Jobs (Processi).
 - c. Scegli Aggiungi lavoro, quindi scegli il lavoro che desideri eseguire.
 - d. Sul Azione scheda, scegli Esegui lavoro.

Componenti SSIS che AWS SCT può convertire in AWS Glue

Puoi usare AWS SCT per convertire i componenti del flusso di dati e del flusso di controllo, nonché i contenitori, i parametri e le variabili.

I componenti del flusso di dati supportati includono quanto segue:

- Destinazione ADO NET
- Fonte ADO NET
- Aggregazione
- Trasformazione della cache
- Trasformazione della mappa dei personaggi
- Trasformazione divisa condizionale
- Copia la trasformazione delle colonne
- Trasformazione della conversione dei dati
- Trasformazione derivata delle colonne
- Destinazione Excel
- Fonte Excel
- Trasformazione delle colonne di esportazione
- Destinazione file flat
- Sorgente file flat
- Trasformazione di Fuzzy Lookup
- Trasformazione delle colonne di importazione

- Trasformazione della ricerca
- Unisci e unisci trasformazione
- Unisci trasformazione
- Trasformazione multicast
- Destinazione ODBC
- Fonte ODBC
- Trasformazione dei comandi OLE DB
- Destinazione OLE DB
- Fonte OLE DB
- Trasformazione percentuale del campionamento
- Trasformazione a perno
- Destinazione del file non elaborato
- Origine file non elaborata
- RecordSetDestinazione
- Trasformazione del numero di righe
- Trasformazione del campionamento di righe
- Ordina trasformazione
- Destinazione SQL Server
- Union All Transformation
- Trasformazione Unpivot
- Fonte XML

I componenti del flusso di controllo supportati includono:

- Attività di inserimento in blocco
- Esecuzione di attività del pacchetto
- Esecuzione di attività SQL
- Esecuzione di attività istruzione T-SQL
- Attività di espressione
- Attività del file system
- Notifica di attività operatore

- Attività di invio di posta

I contenitori SSIS supportati includono:

- Per Loop Container
- Per ogni contenitore Loop
- Contenitore di sequenza

Conversione di SSIS inAWS Glue StudioconAWS SCT

Puoi usareAWS SCTper convertire i pacchetti Microsoft SQL Server Integration Services (SSIS) inAWS Glue Studio.

UnPacchetto SSISinclude i componenti necessari, come il gestore della connessione, le attività, il flusso di controllo, il flusso di dati, i parametri, i gestori di eventi e le variabili, per eseguire una specifica attività di estrazione, trasformazione e caricamento (ETL).AWS SCTconverte i pacchetti SSIS in un formato compatibile conAWS Glue Studio. Dopo aver eseguito la migrazione del database di origine alCloud AWS, puoi eseguire questi convertitiAWS Glue Studiolavori per eseguire attività ETL.

Per convertire i pacchetti Microsoft SSIS inAWS Glue Studio, assicurati di utilizzareAWS SCTversione 1.0.661 o superiore.

Argomenti

- [Prerequisiti](#)
- [Aggiungere pacchetti SSIS al tuoAWS SCTprogetto](#)
- [Conversione di pacchetti SSIS inAWS Glue StudioconAWS SCT](#)
- [CreandoAWS Glue Studiolavori che utilizzano il codice convertito](#)
- [Creazione di un rapporto di valutazione per un pacchetto SSIS conAWS SCT](#)
- [Componenti SSIS cheAWS SCTpuò convertire inAWS Glue Studio](#)

Prerequisiti

In questa sezione, scopri le attività prerequisite per la conversione dei pacchetti SSIS inAWS Glue. Queste attività includono la creazione di elementi obbligatoriAWSrisorse nel tuo account.

Puoi usare AWS Identity and Access Management (IAM) per definire le politiche e i ruoli necessari per accedere a risorse che AWS Glue Studio usi. Per ulteriori informazioni, vedere [Autorizzazioni IAM per AWS Glue Studio utente](#).

Dopo che AWS SCT converte i tuoi script sorgente in AWS Glue Studio, carica gli script convertiti in un bucket Amazon S3. Assicurati di creare questo bucket Amazon S3 e di selezionarlo nelle impostazioni del profilo di servizio. Per ulteriori informazioni sulla creazione di un bucket S3, consulta [Crea il tuo primo bucket S3](#) nella Guida per l'utente di Amazon Simple Storage Service.

Per fare in modo che AWS Glue Studio può connettersi al tuo archivio dati, creare un connettore personalizzato e una connessione. Inoltre, archivia le credenziali del database in AWS Secrets Manager.

Per creare un connettore personalizzato

1. Scarica il driver JDBC per il tuo archivio dati. Per ulteriori informazioni sui driver JDBC AWS SCT usi, vedi [Scaricamento dei driver di database richiesti](#).
 2. Carica questo file del driver nel tuo bucket Amazon S3. Per ulteriori informazioni, vedere [Carica un oggetto nel tuo bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.
 3. Accedi alla AWS Management Console e apri la console AWS Glue Studio all'indirizzo <https://console.aws.amazon.com/gluestudio/>.
 4. Scegli Connettori, quindi scegli Crea un connettore personalizzato.
 5. Per Connettore S3 URL, scegli Sfoglia S3 e scegli il file del driver JDBC che hai caricato nel tuo bucket Amazon S3.
 6. Inserisci una descrizione Nome per il tuo connettore. Ad esempio, specifica **SQLServer**.
 7. Per Tipo di connettore, scegli JDBC.
 8. Per Nome della classe, inserisci il nome della classe principale per il tuo driver JDBC. Per SQL Server, immettere **com.microsoft.sqlserver.jdbc.SQLServerDriver**.
 9. Per Base URL JDBC, inserisci l'URL di base JDBC. La sintassi dell'URL di base JDBC dipende dal motore del database di origine. Per SQL Server, utilizzare il seguente formato: **jdbc:sqlserver://\$<host>:\$<port>;databaseName=\$<dbname>;user=\$<username>;password=\$<password>**.
- Assicurati di sostituire *<host>*, *<port>*, *<dbname>*, *<username>*, e *<password>* con i tuoi valori.
10. Per delimitatore di parametri URL, inserisci il punto e virgola (;).
 11. Scegli Create connector (Crea connettore).

Per archiviare le credenziali del database inAWS Secrets Manager

1. Accedi alAWS Management Consolee apri ilAWS Secrets Managerconsole a<https://console.aws.amazon.com/secretsmanager/>.
2. Scegli Archivia un nuovo segreto.
3. Nella pagina Choose secret type (Scegli il tipo di segreto), effettua le seguenti operazioni:
 - a. PerTipo segreto, scegli laAltro tipo di segreto.
 - b. PerCoppie chiave/valore, inserisci i seguenti tasti:**host,port,dbname,username, epassword**.

Quindi, inserisci i valori per queste chiavi.

4. SulConfigura segretopagina, inserisci una descrizioneNome segreto. Ad esempio, specifica **SQL_Server_secret**.
5. Seleziona Successivo. Quindi, sulConfigura la rotazionepagina, scegliProssimoancora.
6. Nella pagina Review (Revisione), rivedi i dettagli dei segreti e quindi scegli Store (Archivia).

Per creare una connessione per il connettore

1. Accedi alla AWS Management Console e apri la console AWS Glue Studio all'indirizzo <https://console.aws.amazon.com/gluestudio/>.
2. Scegli il connettore per il quale desideri creare una connessione, quindi scegliCrea connessione.
3. SulCrea connessionepagina, inserisci una descrizioneNomeper la tua connessione. Ad esempio, specifica **SQL-Server-connection**.
4. PerAWS Segreto, scegli il segreto che hai creato inAWS Secrets Manager.
5. ConfigurareOpzioni di rete, quindi scegliCrea connessione.

Ora puoi creare unAWS Glue Studiofunziona con un connettore personalizzato. Per ulteriori informazioni, consulta [CreandoAWS Glue Studiolavori](#).

Aggiungere pacchetti SSIS al tuoAWS SCTprogetto

È possibile aggiungere più pacchetti SSIS a uno soloAWS SCTprogetto.

Per aggiungere un pacchetto SSIS al tuo AWS SCT progetto

1. Crea un nuovo progetto in AWS SCT o apri un progetto esistente. Per ulteriori informazioni, consulta [the section called “Creazione di un progetto”](#).
2. Scegli **Aggiungi fonte** dal menu, quindi scegli **Servizi di integrazione SQL Server**.
3. Per **Nome della connessione**, inserisci un nome per i tuoi pacchetti SSIS. AWS SCT visualizza questo nome nell'albero nel pannello di sinistra.
4. Per **cartella dei pacchetti SSIS**, inserisci il percorso della cartella con i pacchetti SSIS di origine.
5. Scegli **Aggiungi obiettivo** dal menu, quindi scegli **AWS Glue Studio**.

Per connettersi a AWS Glue Studio, AWS SCT usa il tuo AWS profilo. Per ulteriori informazioni, consulta [Archiviazione dei profili AWS di servizio in AWS SCT](#).

6. Crea una regola di mappatura, che includa il tuo pacchetto SSIS di origine e il tuo AWS Glue Studio bersaglio. Per ulteriori informazioni, consulta [Creazione di regole di mappatura in AWS SCT](#).
7. Crea AWS Glue Studio connessioni nel AWS Glue Studio console. Per ulteriori informazioni, vedere [Creazione di connessioni per connettori](#).
8. Scegli **Gestori delle connessioni** nell'albero a sinistra, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli **Configurazione delle connessioni**.

AWS SCT visualizza la **Configurazione delle connessioni** in finestra.

9. Per ogni connessione SSIS di origine, scegli un AWS Glue Studio connessione.

Conversione di pacchetti SSIS in AWS Glue Studio con AWS SCT

Di seguito, scopri come convertire i pacchetti SSIS in AWS Glue Studio utilizzando AWS SCT.

Per convertire un pacchetto SSIS in AWS Glue Studio

1. Aggiungi il tuo pacchetto SSIS al tuo AWS SCT progetto. Per ulteriori informazioni, consulta [Aggiungere pacchetti SSIS al tuo AWS SCT progetto](#).
2. Nel pannello di sinistra, espandi **ETL e SSIS nodi**.
3. Scegli **Pacchetti**, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli **Converti pacchetto**.

AWS SCT converte i pacchetti SSIS selezionati in file JSON. Questi oggetti JSON rappresentano un nodo in un grafo aciclico diretto (DAG). Trova i tuoi file convertiti nel Pacchetto DAG nodo nell'albero giusto.

4. Scegli Pacchetto DAGs, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli Salva su Amazon S3.

Ora puoi usare questi script per creare lavori in AWS Glue Studio.

Creando AWS Glue Studio lavori che utilizzano il codice convertito

Dopo aver convertito i pacchetti SSIS di origine, puoi utilizzare i file JSON convertiti per creare AWS Glue Studio lavori.

Per creare un AWS Glue Studio occupazione

1. Scegli Pacchetto DAG nell'albero a destra, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli Configurare AWS Glue Studio occupazione.
2. (Facoltativo) Applica il pacchetto di estensione che emula le funzioni SSIS in AWS Glue Studio.
3. La Configurare AWS Glue Studio occupazione si apre una finestra.

Completa il Proprietà lavorative di base sezione:

- Nome— Inserisci il tuo nome AWS Glue Studio lavoro.
- Nome del file di script— Inserisci un nome per il tuo job script.
- Parametri del lavoro— Aggiungere parametri e inserire i relativi valori.

Seleziona Successivo.

4. Completa il Proprietà lavorative avanzate sezione:

- Ruolo IAM— Scegli il ruolo IAM utilizzato per l'autorizzazione a AWS Glue Studio e accedi agli archivi dati.
- Percorso S3 del file di script— Inserisci il percorso Amazon S3 dello script convertito.
- Elenco temporaneo— Inserisci il percorso Amazon S3 in una directory temporanea per risultati intermedi. AWS Glue Studio utilizza questa directory per leggere o scrivere su Amazon Redshift.

- AWS SCT genera automaticamente il percorso per le librerie Python. Puoi rivedere questo percorso in Percorso della libreria python generato. Non puoi modificare questo percorso generato automaticamente. Per utilizzare librerie Python aggiuntive, inserisci il percorso in Percorso della libreria python dell'utente.
- Percorso della libreria python dell'utente— Inserisci i percorsi per le librerie Python aggiuntive degli utenti. Separa i percorsi Amazon S3 con una virgola.
- Percorso dipendente dei vasi— Inserisci i percorsi per dipendenti* . jar fascicoli. Separa i percorsi Amazon S3 con una virgola.
- Percorso dei file di riferimento— Inserisci i percorsi per i file aggiuntivi, come i file di configurazione, richiesti dallo script. Separa i percorsi Amazon S3 con una virgola.
- Tipo di lavoratore— Scegli G . 1X o G . 2X.

Quando scegli G . 1X ogni worker viene mappato su 1 DPU (4 vCPU, 16 GB di memoria e 64 GB di disco).

Quando scegli G . 2X ogni worker è mappato a 2 DPU (8 vCPU, 32 GB di memoria e 128 GB di disco).

- Numero di lavoratori richiesto— Inserire il numero di lavoratori assegnati durante l'esecuzione del processo.
- Concorrenza massima— Inserire il numero massimo di esecuzioni simultanee consentite per questo processo. L'impostazione predefinita è 1. AWS Glue restituisce un errore quando viene raggiunta questa soglia.
- Timeout del lavoro (minuti)— Inserisci il valore di timeout sul tuo processo ETL come protezione contro i lavori fuori produzione. L'impostazione predefinita è 2.880 minuti (48 ore) per i processi batch. Se il processo supera questo limite, lo stato di esecuzione del processo cambia in TIMEOUT.
- Soglia di notifica di ritardo (minuti)— Inserisci la soglia pochi minuti prima AWS SCT invia una notifica di ritardo.
- Numero di tentativi— Inserisci il numero di volte (0-10) che AWS Glue dovrebbe riavviare automaticamente il processo in caso di errore. I lavori che raggiungono il limite di timeout non vengono riavviati. Il valore predefinito è 0.

Scegli Finish (Fine).

AWS SCT configura il tuo selezionato AWS Glue Studio lavori.

5. Trova i lavori configurati sotto Offerte di lavoro ETL nell'albero giusto. Scegli il lavoro configurato, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli Crea AWS Glue Studio occupazione.
6. Scegli Applica lo stato e assicurati che lo stato il valore per il tuo lavoro è Successo.
7. Apri il AWS Glue Studio console, scegli Aggiorna e scegli il tuo lavoro. Quindi scegli Esegui.

Creazione di un rapporto di valutazione per un pacchetto SSIS con AWS SCT

La Rapporto di valutazione della migrazione ETL fornisce informazioni sulla conversione dei pacchetti SSIS in un formato compatibile con AWS Glue Studio. Il rapporto di valutazione include le azioni relative ai componenti dei pacchetti SSIS. Queste azioni mostrano quali componenti AWS SCT non è possibile eseguire la conversione automatica.

Per creare un rapporto di valutazione della migrazione ETL

1. Espandi il SSIS nodo sotto ETL nel pannello di sinistra.
2. Scegli Pacchetti, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli Crea rapporto.
3. Visualizza la Riepilogo linguetta. Ecco, AWS SCT visualizza le informazioni riepilogative del rapporto di valutazione della migrazione ETL. Include i risultati di conversione per tutti i componenti dei pacchetti SSIS.
4. (Facoltativo) Salvate una copia locale del rapporto di valutazione della migrazione ETL come file PDF o come file con valori separati da virgole (CSV):

- Per salvare il rapporto di valutazione della migrazione ETL come file PDF, scegli Salva in PDF in alto a destra.

Il file PDF contiene il riepilogo esecutivo, le azioni e i consigli per la conversione degli script.

- Per salvare il rapporto di valutazione della migrazione ETL come file CSV, scegli Salva in formato CSV in alto a destra.

AWS SCT crea tre file CSV. Questi file contengono azioni, azioni consigliate e una stima della complessità degli sforzi manuali necessari per convertire gli script.

5. Scegli gli oggetti d'azione. Questa scheda contiene un elenco di elementi che richiedono la conversione manuale in AWS Glue Studio. Quando scegli un'azione dall'elenco, AWS SCT evidenzia l'elemento del pacchetto SSIS di origine a cui si applica l'azione.

Componenti SSIS che AWS SCT può convertire in AWS Glue Studio

Puoi usare AWS SCT per convertire i componenti e i parametri del flusso di dati SSIS in AWS Glue Studio.

I componenti del flusso di dati supportati includono quanto segue:

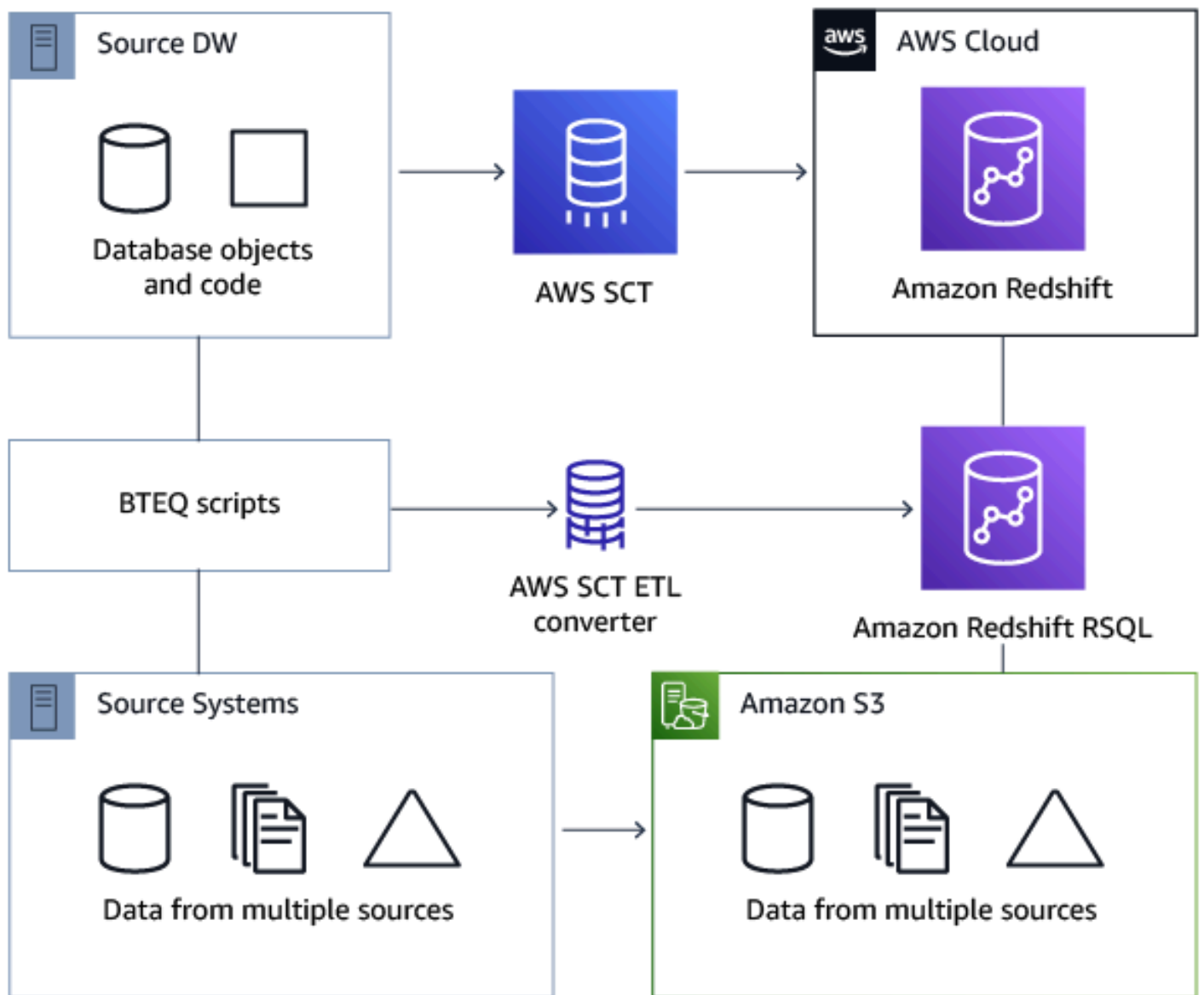
- Destinazione ADO NET
- Fonte ADO NET
- Aggregazione
- Mappa dei personaggi
- Divisione condizionale
- Copia colonna
- Conversione dei dati
- Colonna derivata
- Cercare
- Unione
- Merge join
- Multicast
- Destinazione ODBC
- Fonte ODBC
- Destinazione OLEDB
- Fonte OLEDB
- Numero di righe
- Ordina
- Destinazione SQL Server
- Union All

AWS SCT può convertire più componenti SSIS in AWS Glue. Per ulteriori informazioni, consulta [Componenti SSIS che AWS SCT può convertire in AWS Glue](#).

Conversione degli script Teradata BTEQ in Amazon Redshift RSQL con AWS SCT

Puoi usare il AWS Schema Conversion Tool (AWS SCT) per convertire gli script Teradata Basic Teradata Query (BTEQ) in Amazon Redshift RSQL.

Il seguente diagramma di architettura mostra il progetto di migrazione del database che include la conversione degli script di estrazione, trasformazione e caricamento (ETL) in Amazon Redshift RSQL.



Argomenti

- [Aggiungere script BTEQ al tuo AWS SCT progetto](#)
- [Configurazione delle variabili di sostituzione negli script BTEQ con AWS SCT](#)
- [Conversione degli script Teradata BTEQ in Amazon Redshift RSQL con AWS SCT](#)
- [Gestione degli script BTEQ con AWS SCT](#)
- [Creazione di un rapporto di valutazione della conversione degli script BTEQ con AWS SCT](#)
- [Modifica e salvataggio degli script BTEQ convertiti con AWS SCT](#)

Aggiungere script BTEQ al tuo AWS SCT progetto

È possibile aggiungere più script a uno solo AWS SCT progetto.

Per aggiungere uno script BTEQ al tuo AWS SCT progetto

1. Crea un nuovo progetto in AWS SCT o apri un progetto esistente. Per ulteriori informazioni, consulta [the section called “Creazione di un progetto”](#).
2. Scegli **Aggiungi fonte** dal menu, quindi scegli **Teradata** per aggiungere il tuo database di origine al progetto. Per ulteriori informazioni, consulta [Utilizzo di Teradata come fonte](#).
3. Scegli **Aggiungi obiettivo** dal menu per aggiungere un database Amazon Redshift di destinazione al tuo AWS SCT progetto.

Puoi utilizzare una piattaforma di database di destinazione Amazon Redshift virtuale. Per ulteriori informazioni, consulta [Utilizzo di obiettivi virtuali](#).

4. Crea una nuova regola di mappatura che includa il tuo database Teradata di origine e il tuo target Amazon Redshift. Per ulteriori informazioni, consulta [Aggiungere una nuova regola di mappatura](#).
5. Sul **Visualizza** menu, scegli **Vista principale**.
6. Nel pannello di sinistra, espandi **Script nodo**.
7. Scegli **Script BTEQ**, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli **Caricare gli script**.
8. Inserisci la posizione del codice sorgente per i tuoi script Teradata BTEQ e scegli **Seleziona cartella**.

AWS SCT visualizza il **Caricare gli script** finestra.

9. Completa una delle seguenti operazioni:
 - a. Se gli script Teradata BTEQ non includono le variabili di sostituzione, scegli **Nessuna variabile di sostituzione**, quindi scegli **OK** per aggiungere script al tuo AWS SCT progetto.
 - b. Se gli script Teradata BTEQ includono le variabili di sostituzione, configura le variabili di sostituzione. Per ulteriori informazioni, consulta [Configurazione delle variabili di sostituzione negli script BTEQ](#).

Configurazione delle variabili di sostituzione negli script BTEQ conAWS SCT

Gli script Teradata BTEQ possono includere variabili di sostituzione. Ad esempio, puoi utilizzare uno script BTEQ con variabili di sostituzione per eseguire lo stesso set di comandi su più ambienti di database. Puoi usareAWS SCTper configurare le variabili di sostituzione negli script BTEQ.

Prima di eseguire uno script BTEQ con variabili di sostituzione, assicurati di assegnare i valori per tutte le variabili. Per fare ciò, puoi usare altri strumenti o applicazioni come uno script Bash, UC4 (Automatic) e così via.AWS SCTpuò risolvere e convertire le variabili di sostituzione solo dopo averne assegnato i valori.

Per configurare le variabili di sostituzione nello script BTEQ

1. Aggiungi i tuoi script BTEQ al tuoAWS SCTprogetto. Per ulteriori informazioni, consulta [Aggiungere script BTEQ al tuoAWS SCTprogetto](#).

Quando aggiungi i tuoi script, scegliVengono utilizzate variabili di sostituzione.

2. PerDefinire il formato variabile, inserisci un'espressione regolare che corrisponda a tutte le variabili di sostituzione nello script.

Ad esempio, se i nomi delle variabili di sostituzione iniziano con\${e termina con}, usa`\$\{\w+ \}`espressione regolare. Per abbinare le variabili di sostituzione che iniziano con il simbolo del dollaro o del simbolo di percentuale, utilizzate il`\$\w+ | \% \w+`espressione regolare.

Le espressioni regolari in AWS SCT sono conformi alla sintassi delle espressioni regolari Java. Per ulteriori informazioni, vedere[Modello di classe java.util.regex](#)nella documentazione Java.

3. ScegliOKper caricare script sul tuoAWS SCTprogetto, quindi scegliOKper chiudere ilCaricare gli scriptfinestra.
4. ScegliVariabiliper visualizzare tutte le variabili di sostituzione scoperte e i relativi valori.
5. PerValore, inserite il valore per la variabile di sostituzione.

Conversione degli script Teradata BTEQ in Amazon Redshift RSQL conAWS SCT

Di seguito, scopri come convertire gli script ETL BTEQ in Amazon Redshift RSQL utilizzandoAWS SCT.

Per convertire uno script Teradata BTEQ in Amazon Redshift RSQL

1. Aggiungi i tuoi script BTEQ al tuoAWS SCTprogetto. Per ulteriori informazioni, consulta [Aggiungere script BTEQ al tuoAWS SCTprogetto](#).
2. Configura le variabili di sostituzione. Per ulteriori informazioni, consulta [Configurazione delle variabili di sostituzione negli script BTEQ](#).
3. Nel pannello di sinistra, espandiScriptnodo.
4. Completa una delle seguenti operazioni:
 - Per convertire un singolo script BTEQ, espandiScript BTEQnodo, scegli lo script da convertire, quindi scegliConverti in RSQLdal menu contestuale (tasto destro).
 - Per nascondere più script, assicurati di aver selezionato tutti gli script da convertire. Quindi scegliScript BTEQ, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegliConverti in RSQLsottoConverti script.

AWS SCTconverte tutti gli script Teradata BTEQ selezionati in un formato compatibile con Amazon Redshift RSQL. Trova i tuoi script convertiti nelScriptnodo nel pannello del database di destinazione.

5. Modifica gli script RSQL di Amazon Redshift convertiti o salvati. Per ulteriori informazioni, consulta [Modifica e salvataggio degli script BTEQ convertiti](#).

Gestione degli script BTEQ conAWS SCT

Puoi aggiungere più script BTEQ o rimuovere uno script BTEQ dal tuoAWS SCTprogetto.

Per aggiungere uno script BTEQ aggiuntivo al tuoAWS SCTprogetto

1. Espandi ilScriptnodo nel pannello di sinistra.
2. Scegli laScript BTEQnodo e apri il menu contestuale (clic con il pulsante destro del mouse).
3. ScegliCaricare gli script.
4. Inserisci le informazioni necessarie per aggiungere un nuovo script BTEQ e configurare le variabili di sostituzione. Per ulteriori informazioni, consultare [Aggiungere script BTEQ al tuoAWS SCTprogetto](#) e [Configurazione delle variabili di sostituzione negli script BTEQ](#).

Per rimuovere uno script BTEQ dal tuo AWS SCT progetto

1. Espandi il **Script BTEQ** nodo sotto **Script** nel pannello di sinistra.
2. Scegli lo script da rimuovere e apri il menu contestuale (clic con il pulsante destro del mouse).
3. Scegli **Elimina script**.

Creazione di un rapporto di valutazione della conversione degli script BTEQ con AWS SCT

Un rapporto di valutazione della conversione degli script BTEQ fornisce informazioni sulla conversione dei comandi BTEQ e delle istruzioni SQL dagli script BTEQ in un formato compatibile con Amazon Redshift RSQL. Il rapporto di valutazione include azioni per i comandi BTEQ e le istruzioni SQL che AWS SCT impossibile convertire.

Per creare un rapporto di valutazione della conversione degli script BTEQ

1. Espandi il **Script BTEQ** nodo sotto **Script** nel pannello di sinistra.
2. Scegli lo script da convertire e apri il menu contestuale (clic con il pulsante destro del mouse).
3. Scegli **Conversione in RSQL** sotto **Crea rapporto**.
4. Visualizza la **Riepilogo** linguetta. La **Riepilogo** scheda mostra le informazioni di riepilogo del rapporto di valutazione dello script BTEQ. Include i risultati di conversione per tutti i comandi BTEQ e le istruzioni SQL degli script BTEQ.
5. (Facoltativo) Salva una copia locale del rapporto di valutazione della conversione dello script BTEQ come file PDF o come file con valori separati da virgole (CSV):
 - Per salvare il rapporto di valutazione della conversione dello script BTEQ come file PDF, scegli **Salva in PDF** in alto a destra.

Il file PDF contiene il riepilogo esecutivo, le azioni e i consigli per la conversione degli script.

- Per salvare il rapporto di valutazione della conversione dello script BTEQ come file CSV, scegli **Salva in formato CSV** in alto a destra.

Il file CSV contiene le azioni, le azioni consigliate e una stima della complessità degli sforzi manuali necessari per convertire gli script.

6. Scegli laOggetti d'azione in lingua. Questa scheda contiene un elenco di articoli che richiedono la conversione manuale in Amazon Redshift RSQL. Quando scegli un'azione dall'elenco,AWS SCT evidenzia l'elemento dello script BTEQ di origine a cui si applica l'azione.

Modifica e salvataggio degli script BTEQ convertiti conAWS SCT

Puoi modificare gli script convertiti nel pannello inferiore del tuoAWS SCTprogetto.AWS SCTmemorizza lo script modificato come parte del progetto.

Per salvare gli script convertiti

1. Espandi ilScript RSQLnodo sottoScriptnel pannello del database di destinazione.
2. Scegli lo script convertito, apri il menu contestuale (clic con il pulsante destro del mouse) e scegliSalva script.
3. Inserisci il percorso della cartella per salvare lo script convertito e scegliSalva.

AWS SCTsalva lo script convertito in un file e apre questo file.

Conversione di script di shell con comandi Teradata BTEQ incorporati in Amazon Redshift RSQL conAWS SCT

Puoi usare ilAWS Schema Conversion Tool(AWS SCT) per convertire script di shell con comandi Teradata Basic Teradata Query (BTEQ) incorporati in script di shell con comandi Amazon Redshift RSQL incorporati.

AWS SCTestrae i comandi Teradata BTEQ dagli script della shell e li converte in un formato compatibile con Amazon Redshift. Dopo la migrazione del database Teradata su Amazon Redshift, puoi utilizzare questi script convertiti per gestire il tuo nuovo database Amazon Redshift.

Puoi anche usareAWS SCTper convertire file con script Teradata BTEQ ETL in Amazon Redshift RSQL. Per ulteriori informazioni, consulta [Conversione degli script Teradata BTEQ in Amazon Redshift RSQL conAWS SCT](#).

Argomenti

- [Aggiungere script di shell con comandi Teradata BTEQ incorporati al tuoAWS SCTprogetto](#)
- [Configurazione delle variabili di sostituzione negli script di shell con comandi Teradata BTEQ incorporati conAWS SCT](#)

- [Conversione di script di shell con comandi Teradata BTEQ incorporati conAWS SCT](#)
- [Gestione degli script di shell con comandi Teradata BTEQ incorporati conAWS SCT](#)
- [Creazione di un rapporto di valutazione per una conversione di script di shell conAWS SCT](#)
- [Modifica e salvataggio degli script di shell convertiti conAWS SCT](#)

Aggiungere script di shell con comandi Teradata BTEQ incorporati al tuoAWS SCTprogetto

È possibile aggiungere più script a uno soloAWS SCTprogetto.

Per aggiungere uno script di shell al tuoAWS SCTprogetto

1. Crea un nuovo progetto inAWS SCTo apri un progetto esistente. Per ulteriori informazioni, consulta [the section called “Creazione di un progetto”](#).
2. ScegliAggiungi fontedal menu, quindi scegliTeradataper aggiungere il tuo database di origine al progetto. Per ulteriori informazioni, consulta [Utilizzo di Teradata come fonte](#).
3. ScegliAggiungi obiettivodal menu e per aggiungere un database Amazon Redshift di destinazione al tuoAWS SCTprogetto.

Puoi utilizzare una piattaforma di database di destinazione Amazon Redshift virtuale. Per ulteriori informazioni, consulta [Utilizzo di obiettivi virtuali](#).

4. Crea una nuova regola di mappatura che includa il tuo database Teradata di origine e il tuo target Amazon Redshift. Per ulteriori informazioni, consulta [Aggiungere una nuova regola di mappatura](#).
5. SulVisualizzamenu, scegliVista principale.
6. Nel pannello di sinistra, espandiScriptnodo.
7. ScegliConchiglia, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegliCaricare gli script.
8. Inserisci la posizione degli script della shell di origine con i comandi Teradata BTEQ incorporati e scegliSeleziona cartella.

AWS SCTvisualizza ilCaricare gli scriptfinestra.

9. Completa una delle seguenti operazioni:

- Se i tuoi script di shell non includono le variabili di sostituzione, scegli **Nessuna variabile di sostituzione**, quindi scegli **OK** per aggiungere script al tuo **AWS SCT** progetto.
- Se i tuoi script di shell includono le variabili di sostituzione, configura le variabili di sostituzione. Per ulteriori informazioni, consulta [Configurazione delle variabili di sostituzione negli script di shell](#).

Configurazione delle variabili di sostituzione negli script di shell con comandi Teradata BTEQ incorporati con **AWS SCT**

I tuoi script di shell possono includere variabili di sostituzione. Ad esempio, è possibile utilizzare un singolo script con variabili di sostituzione per gestire database in ambienti diversi. Puoi usare **AWS SCT** per configurare le variabili di sostituzione negli script di shell.

Prima di eseguire i comandi BTEQ con variabili di sostituzione da uno script di shell, assicuratevi di assegnare i valori per tutte le variabili all'interno di questo script di shell. **AWS SCT** può risolvere e convertire le variabili di sostituzione solo dopo averne assegnato i valori.

Per configurare le variabili di sostituzione nello script di shell

1. Aggiungi i tuoi script di shell sorgente al tuo **AWS SCT** progetto. Per ulteriori informazioni, consulta [Aggiungere script di shell al tuo **AWS SCT** progetto](#).

Quando aggiungi i tuoi script, scegli **Vengono utilizzate variabili di sostituzione**.

2. Per **Definire il formato variabile**, inserisci un'espressione regolare che corrisponda a tutte le variabili di sostituzione nello script.

Ad esempio, se i nomi delle variabili di sostituzione iniziano con `{e termina con}`, usa `\$\{\w+ \}` espressione regolare. Per abbinare le variabili di sostituzione che iniziano con il simbolo del dollaro o con il simbolo di percentuale, usa `\$\w+ | \% \w+` espressione regolare.

Le espressioni regolari in **AWS SCT** sono conformi alla sintassi delle espressioni regolari Java. Per ulteriori informazioni, vedere [Modello di classe java.util.regex](#) nella documentazione Java.

3. Scegli **OK** per caricare script sul tuo **AWS SCT** progetto, quindi scegli **OK** per chiudere il **Caricare gli script** finestra.
4. Scegli **Variabili** per visualizzare tutte le variabili di sostituzione scoperte e i relativi valori.
5. Per **Valore**, inserite il valore per la variabile di sostituzione.

Conversione di script di shell con comandi Teradata BTEQ incorporati conAWS SCT

Di seguito, scopri come convertire script di shell con comandi Teradata BTEQ incorporati in script di shell con comandi Amazon Redshift RSQL incorporati utilizzandoAWS SCT.

Per convertire uno script di shell

1. Aggiungi i tuoi script di shell al tuoAWS SCTprogetto. Per ulteriori informazioni, consulta [Aggiungere script di shell al tuoAWS SCTprogetto](#).
2. Configura le variabili di sostituzione. Per ulteriori informazioni, consulta [Configurazione delle variabili di sostituzione negli script di shell](#).
3. Nel pannello di sinistra, espandiScriptnodo.
4. Completa una delle seguenti operazioni:
 - Per convertire i comandi BTEQ da un singolo script di shell, espandiConchiglianodo, scegli lo script da convertire, quindi scegliConverti scriptdal menu contestuale (tasto destro).
 - Per nascondere più script, assicurati di aver selezionato tutti gli script da convertire. Quindi scegliConchiglia, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegliConverti script.
5. Scegli OK.

AWS SCTconverte i comandi BTEQ negli script di shell selezionati in un formato compatibile con Amazon Redshift RSQL. Trova i tuoi script convertiti nelScriptnodo nel pannello del database di destinazione.

6. Modifica gli script RSQL di Amazon Redshift convertiti o salvati. Per ulteriori informazioni, consulta [Modifica e salvataggio degli script di shell convertiti](#).

Gestione degli script di shell con comandi Teradata BTEQ incorporati conAWS SCT

Puoi aggiungere più script di shell o rimuovere uno script di shell dal tuoAWS SCTprogetto.

Per aggiungere un nuovo script di shell al tuoAWS SCTprogetto

1. Espandi ilScriptnodo nel pannello di sinistra.

2. Scegli **laConchiglianodo** e apri il menu contestuale (clic con il pulsante destro del mouse).
3. Scegli **Caricare** gli script.
4. Immettete le informazioni necessarie per aggiungere un nuovo script di shell e configurare le variabili di sostituzione. Per ulteriori informazioni, consultate [Aggiungere script di shell al tuoAWS SCTprogetto](#) e [Configurazione delle variabili di sostituzione negli script di shell](#).

Per rimuovere uno script di shell dal tuoAWS SCTprogetto

1. Espandi il **Conchiglianodo** sotto **Script** nel pannello di sinistra.
2. Scegli lo script da rimuovere e apri il menu contestuale (clic con il pulsante destro del mouse).
3. Scegli **Elimina** script.

Creazione di un rapporto di valutazione per una conversione di script di shell conAWS SCT

La **rapporto di valutazione** della conversione degli script di shell fornisce informazioni sulla conversione dei comandi BTEQ e delle istruzioni SQL. La conversione avviene dagli script di origine in un formato compatibile con Amazon Redshift RSQL. Il rapporto di valutazione include azioni per i comandi BTEQ e le istruzioni SQL cheAWS SCTimpossibile convertire.

Per creare un rapporto di valutazione della conversione degli script di shell

1. Espandi il **Conchiglianodo** sotto **Script** nel pannello di sinistra.
2. Scegli lo script da convertire, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli **Crea** rapporto.
3. Visualizza la **Riepilogo** linguetta. La **Riepilogo** scheda mostra le informazioni di riepilogo del rapporto di valutazione dello script di shell. Include i risultati di conversione per tutti i comandi BTEQ e le istruzioni SQL degli script di origine.
4. (Facoltativo) Salvate una copia locale del rapporto di valutazione della conversione dello script di shell come file PDF o come file con valori separati da virgole (CSV):
 - Per salvare il rapporto di valutazione della conversione dello script shell come file PDF, scegliete **Salva in PDF** in alto a destra.

Il file PDF contiene il riepilogo esecutivo, le azioni e i consigli per la conversione degli script.

- Per salvare il rapporto di valutazione della conversione dello script di shell come file CSV, scegli **Salva in formato CSV** in alto a destra.

Il file CSV contiene le azioni, le azioni consigliate e una stima della complessità degli sforzi manuali necessari per convertire gli script.

5. Scegli **LaOggetti d'azione** linguetta. Questa scheda contiene un elenco di articoli che richiedono la conversione manuale in Amazon Redshift RSQL. Quando si seleziona un'azione dall'elenco, AWS SCT evidenzia l'elemento dello script della shell di origine a cui si applica l'azione.

Modifica e salvataggio degli script di shell convertiti con AWS SCT

Puoi modificare gli script convertiti nel pannello inferiore del tuo AWS SCT progetto. AWS SCT memorizza lo script modificato come parte del progetto.

Per salvare gli script convertiti

1. Espandi il **Script RSQL** nodo sotto **Script** nel pannello del database di destinazione.
2. Scegli lo script convertito, apri il menu contestuale (clic con il pulsante destro del mouse) e scegli **Salva script**.
3. Inserisci il percorso della cartella per salvare lo script convertito e scegli **Salva**.

AWS SCT salva lo script convertito in un file e apre questo file.

Conversione di Teradata FastExport script di lavoro su Amazon Redshift RSQL con AWS SCT

Puoi usare il **AWS Schema Conversion Tool (AWS SCT)** per convertire **Teradata FastExport** script di lavoro per Amazon Redshift RSQL.

Un **FastExport** script di lavoro è un set di **FastExport** comandi e istruzioni SQL che selezionano ed esportano dati dal database Teradata. AWS SCT converte **FastExport** comandi e istruzioni SQL in un formato compatibile con Amazon Redshift RSQL. Dopo la migrazione del database Teradata su Amazon Redshift, puoi utilizzare questi script convertiti per esportare dati dal database Amazon Redshift.

Argomenti

- [AggiungendoFastExportscript di lavoro per i tuoiAWS SCTprogetto](#)
- [Configurazione delle variabili di sostituzione in TeradataFastExportscript di lavoro conAWS SCT](#)
- [Conversione di TeradataFastExportscript di lavoro conAWS SCT](#)
- [Gestione di TeradataFastExportscript di lavoro conAWS SCT](#)
- [Creazione di un rapporto di valutazione per un TeradataFastExportconversione di script di lavoro conAWS SCT](#)
- [Modifica e salvataggio dei tuoi Teradata convertitiFastExportscript di lavoro conAWS SCT](#)

AggiungendoFastExportscript di lavoro per i tuoiAWS SCTprogetto

È possibile aggiungere più script a uno soloAWS SCTprogetto.

Per aggiungere unFastExportscript di lavoro per il tuoAWS SCTprogetto

1. Crea un nuovo progetto inAWS SCTo apri un progetto esistente. Per ulteriori informazioni, consulta [the section called “Creazione di un progetto”](#).
2. ScegliAggiungi fontedal menu, quindi scegliTeradataper aggiungere il tuo database di origine al progetto. Per ulteriori informazioni, consulta [Utilizzo di Teradata come fonte](#).
3. ScegliAggiungi obiettivodal menu e per aggiungere un database Amazon Redshift di destinazione al tuoAWS SCTprogetto.

Puoi utilizzare una piattaforma di database di destinazione Amazon Redshift virtuale. Per ulteriori informazioni, consulta [Utilizzo di obiettivi virtuali](#).

4. Crea una nuova regola di mappatura che includa il tuo database Teradata di origine e il tuo target Amazon Redshift. Per ulteriori informazioni, consulta [Aggiungere una nuova regola di mappatura](#).
5. SulVisualizzamenu, scegliVista principale.
6. Nel pannello di sinistra, espandiScriptnodo.
7. ScegliFastExport, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegliCaricare gli script.
8. Inserisci la posizione del codice sorgente del tuo TeradataFastExportscript di lavoro e scegliSeleziona cartella.

AWS SCTvisualizza ilCaricare gli scriptfinestra.

9. Completa una delle seguenti operazioni:

- Se il tuo TeradataFastExportgli script di lavoro non includono le variabili di sostituzione, scegliNessuna variabile di sostituzionee poi scegliOKper aggiungere script al tuoAWS SCTprogetto.
- Se il tuo TeradataFastExportgli script di lavoro includono le variabili di sostituzione, configurano le variabili di sostituzione. Per ulteriori informazioni, consulta [Configurazione delle variabili di sostituzione inFastExportscript di lavoro](#).

Configurazione delle variabili di sostituzione in TeradataFastExportscript di lavoro conAWS SCT

Il tuo TeradataFastExportgli script di lavoro possono includere variabili di sostituzione. Ad esempio, è possibile utilizzare un singolo script con variabili di sostituzione per esportare dati da più database. Puoi usareAWS SCTper configurare le variabili di sostituzione negli script Teradata.

Prima di eseguire unFastExportjob script con variabili di sostituzione, assicurati di assegnare i valori per tutte le variabili. Per fare ciò, puoi usare altri strumenti o applicazioni come uno script Bash, UC4 (Automic) e così via.AWS SCTpuò risolvere e convertire le variabili di sostituzione solo dopo averne assegnato i valori.

Per configurare le variabili di sostituzione nel tuoFastExportscript di lavoro

1. Aggiungi la tua fonte TeradataFastExportscript di lavoro per i tuoiAWS SCTprogetto. Per ulteriori informazioni, consulta [Aggiungere script BTEQ al tuoAWS SCTprogetto](#).

Quando aggiungi i tuoi script, scegliVengono utilizzate variabili di sostituzione.

2. PerDefinire il formato variabile, inserisci un'espressione regolare che corrisponda a tutte le variabili di sostituzione nello script.

Ad esempio, se i nomi delle variabili di sostituzione iniziano con\${e termina con}, usa il\$\{\w + \}espressione regolare. Per abbinare le variabili di sostituzione che iniziano con il simbolo del dollaro o con il simbolo di percentuale, usa\$\w + | \% \w +espressione regolare.

Le espressioni regolari in AWS SCT sono conformi alla sintassi delle espressioni regolari Java. Per ulteriori informazioni, vedere[Modello di classe java.util.regex](#)nella documentazione Java.

3. ScegliOKper caricare script sul tuoAWS SCTprogetto, quindi scegliOKper chiudere ilCaricare gli scriptfinestra.

4. Nel pannello di sinistra, espandiScriptnodo. ScegliFastExport, quindi scegli la tua cartella con gli script. Apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegliEsporta variabilisottoVariabili di sostituzione.
5. Esporta le variabili di sostituzione per uno script. Espandi la cartella con gli script, scegli il tuo script, apri il menu contestuale (clic con il pulsante destro del mouse) e scegliEsporta variabilisottoVariabili di sostituzione.
6. Inserisci il nome del file con valori separati da virgole (CSV) per salvare le variabili di sostituzione e scegliSalva.
7. Apri questo file CSV e inserisci i valori per le variabili di sostituzione.

A seconda del sistema operativo,AWS SCTutilizza diversi formati per i file CSV. I valori nel file possono essere racchiusi tra virgolette o meno. Assicurati di utilizzare lo stesso formato per i valori delle variabili di sostituzione degli altri valori nel file.AWS SCTnon è possibile importare il file CSV con valori in formati diversi.

8. Salva il file CSV.
9. Nel pannello di sinistra, espandiScriptnodo. ScegliFastExport, quindi scegli il tuo copione. Apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegliImporta variabilisottoVariabili di sostituzione.
10. Scegli il tuo file CSV, quindi scegliAperto.
11. ScegliVariabiliper visualizzare tutte le variabili di sostituzione scoperte e i relativi valori.

Conversione di TeradataFastExportscript di lavoro conAWS SCT

Di seguito, scopri come convertire TeradataFastExportlavoro su Amazon Redshift RSQL utilizzandoAWS SCT.

Per convertire un TeradataFastExportscript di lavoro per Amazon Redshift RSQL

1. Aggiungi il tuoFastExportscript di lavoro per i tuoiAWS SCTprogetto. Per ulteriori informazioni, consulta [AggiungendoFastExportscript di lavoro per i tuoiAWS SCTprogetto](#).
2. Configura le variabili di sostituzione. Per ulteriori informazioni, consulta [Configurazione delle variabili di sostituzione inFastExportscript di lavoro](#).
3. Nel pannello di sinistra, espandiScriptnodo.
4. Completa una delle seguenti operazioni:

- Per convertire un singolo `FastExportjob` script, espandi `FastExportnodo`, scegli lo script da convertire, quindi scegli `Converti script` dal menu contestuale (tasto destro).
- Per nascondere più script, assicurati di aver selezionato tutti gli script da convertire. Quindi scegli `FastExport`, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli `Converti script`.

AWS SCT converte tutti i Teradata selezionati `FastExportscript` di lavoro in un formato compatibile con Amazon Redshift RSQL. Trova i tuoi script convertiti nel `Scriptnodo` nel pannello del database di destinazione.

5. Modifica gli script RSQL di Amazon Redshift convertiti o salvati. Per ulteriori informazioni, consulta [Modifica e salvataggio dei file convertiti `FastExportscript` di lavoro](#).

Gestione di Teradata `FastExportscript` di lavoro con AWS SCT

Puoi aggiungere più Teradata `FastExportscript` di lavoro o rimuovi un `FastExportscript` di lavoro dal tuo AWS SCT progetto.

Per aggiungere un nuovo `FastExportscript` di lavoro per il tuo AWS SCT progetto

1. Espandi il `Scriptnodo` nel pannello di sinistra.
2. Scegli la `FastExportnodo` e apri il menu contestuale (clic con il pulsante destro del mouse).
3. Scegli `Caricare gli script`.
4. Inserisci le informazioni necessarie per aggiungerne una nuova `FastExportscript` di lavoro e configurazione delle variabili di sostituzione. Per ulteriori informazioni, consultare [Aggiungendo `FastExportscript` di lavoro per i tuoi AWS SCT progetto](#) e [Configurazione delle variabili di sostituzione in `FastExportscript` di lavoro](#).

Per rimuovere un `FastExportscript` di lavoro dal tuo AWS SCT progetto

1. Espandi il `FastExportnodo` sotto `Script` nel pannello di sinistra.
2. Scegli lo script da rimuovere e apri il menu contestuale (clic con il pulsante destro del mouse).
3. Scegli `Elimina script`.

Creazione di un rapporto di valutazione per un TeradataFastExportconversione di script di lavoro conAWS SCT

LaFastExportrapporto di valutazione della conversione degli script di lavoro fornisce informazioni sulla conversione diFastExportcomandi e istruzioni SQL dal tuoFastExportscript in un formato compatibile con Amazon Redshift RSQL. Il rapporto di valutazione include elementi di azione perFastExportcomandi e istruzioni SQL cheAWS SCTimpossibile convertire.

Per creare un rapporto di valutazione della conversione degli script per un TeradataFastExportoccupazione

1. Espandi ilFastExportnodo sottoScriptnel pannello di sinistra.
2. Scegli lo script da convertire, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegliCrea rapporto.
3. Visualizza laRiepilogolinguetta. LaRiepilogola scheda mostra le informazioni di riepilogo delFastExportrapporto di valutazione del job script. Include risultati di conversione per tuttiFastExportcomandi e istruzioni SQL dai tuoi script di origine.
4. È possibile salvare una copia locale diFastExportrapporto di valutazione della conversione degli script di lavoro come file PDF o file con valori separati da virgole (CSV).
 - a. Per salvare ilFastExportrapporto di valutazione della conversione degli script di lavoro come file PDF, scegliSalva in PDFin alto a destra.

Il file PDF contiene il riepilogo esecutivo, le azioni e i consigli per la conversione degli script.
 - b. Per salvare ilFastExportrapporto di valutazione della conversione degli script di lavoro come file CSV, scegliSalva in formato CSVin alto a destra.

Il file CSV contiene le azioni, le azioni consigliate e una stima della complessità degli sforzi manuali necessari per convertire gli script.
5. Scegli laOggetti d'azionelinguetta. Questa scheda contiene un elenco di articoli che richiedono la conversione manuale in Amazon Redshift RSQL. Quando si seleziona un'azione dall'elenco,AWS SCTevidenzia l'articolo dalla tua fonteFastExportscript di lavoro a cui si applica l'azione.

Modifica e salvataggio dei tuoi Teradata convertitiFastExportscript di lavoro conAWS SCT

Puoi modificare gli script convertiti nel pannello inferiore del tuoAWS SCTprogetto.AWS SCTmemorizza lo script modificato come parte del progetto.

Per salvare gli script convertiti

1. Espandi ilScript RSQLnodo sottoScriptnel pannello del database di destinazione.
2. Scegli lo script convertito, apri il menu contestuale (clic con il pulsante destro del mouse) e scegliSalva script.
3. Inserisci il percorso della cartella per salvare lo script convertito e scegliSalva.

AWS SCTsalva lo script convertito in un file e apre questo file.

Conversione di TeradataFastLoadscript di lavoro su Amazon Redshift RSQL conAWS SCT

Puoi usare ilAWS Schema Conversion Tool(AWS SCT) per convertire TeradataFastLoadscript di lavoro per Amazon Redshift RSQL.

UNTeradataFastLoadsceneggiaturaè un insieme di comandi che utilizzano più sessioni per caricare dati in una tabella vuota su un database Teradata. TeradataFastLoadelabora una serie di TeradataFastLoadcomandi e istruzioni SQL. Il TeradataFastLoadi comandi forniscono il controllo della sessione e la gestione dei dati trasferiti. Le istruzioni SQL creano, gestiscono ed eliminano tabelle.

AWS SCTconverte TeradataFastLoadcomandi e istruzioni SQL in un formato compatibile con Amazon Redshift RSQL. Dopo la migrazione del database Teradata su Amazon Redshift, puoi utilizzare questi script convertiti per caricare i dati nel tuo database Amazon Redshift.

Argomenti

- [AggiungendoFastLoadscript di lavoro per i tuoiAWS SCTprogetto](#)
- [Configurazione delle variabili di sostituzione in TeradataFastLoadscript di lavoro conAWS SCT](#)
- [Conversione di TeradataFastLoadscript di lavoro conAWS SCT](#)
- [Gestione di TeradataFastLoadscript di lavoro conAWS SCT](#)

- [Creazione di un rapporto di valutazione per un TeradataFastLoadconversione di script di lavoro conAWS SCT](#)
- [Modifica e salvataggio dei tuoi Teradata convertitiFastLoadscript di lavoro conAWS SCT](#)

AggiungendoFastLoadscript di lavoro per i tuoiAWS SCTprogetto

È possibile aggiungere più script a uno soloAWS SCTprogetto.

Per aggiungere unFastLoadscript di lavoro per il tuoAWS SCTprogetto

1. Crea un nuovo progetto inAWS SCToppure apri un progetto esistente. Per ulteriori informazioni, consulta [the section called “Creazione di un progetto”](#).
2. ScegliAggiungi fontedal menu, quindi scegliTeradataper aggiungere il tuo database di origine al progetto. Per ulteriori informazioni, consulta [Utilizzo di Teradata come fonte](#).
3. ScegliAggiungi obiettivodal menu e aggiungi un database Amazon Redshift di destinazione al tuoAWS SCTprogetto.

Puoi utilizzare una piattaforma di database di destinazione Amazon Redshift virtuale. Per ulteriori informazioni, consulta [Utilizzo di obiettivi virtuali](#).

4. Crea una nuova regola di mappatura che includa il tuo database Teradata di origine e il tuo target Amazon Redshift. Per ulteriori informazioni, consulta [Aggiungere una nuova regola di mappatura](#).
5. SulVisualizzamenu, scegliVista principale.
6. Nel pannello di sinistra, espandiScriptnodo.
7. ScegliFastLoad, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegliCaricare gli script.
8. Inserisci la posizione della tua fonte TeradataFastLoadscript di lavoro e scegliSeleziona cartella.

AWS SCTvisualizza ilCaricare gli scriptfinestra.

9. Completa una delle seguenti operazioni:
 - Se il tuo TeradataFastLoadgli script di lavoro non includono le variabili di sostituzione, scegliNessuna variabile di sostituzione, quindi scegliOKper aggiungere script al tuoAWS SCTprogetto.

- Se il tuo TeradataFastLoadgli script di lavoro includono le variabili di sostituzione, configurano le variabili di sostituzione. Per ulteriori informazioni, consulta [Configurazione delle variabili di sostituzione inFastLoadscript di lavoro](#).

Configurazione delle variabili di sostituzione in TeradataFastLoadscript di lavoro conAWS SCT

Il tuo TeradataFastLoadgli script di lavoro potrebbero includere variabili di sostituzione. Ad esempio, è possibile utilizzare un singolo script con variabili di sostituzione per caricare dati su database diversi.

Prima di eseguire unFastLoadjob script con variabili di sostituzione, assicurati di assegnare i valori per tutte le variabili. Per fare ciò, puoi usare altri strumenti o applicazioni come uno script Bash, UC4 (Automic) e così via.

AWS SCTpuò risolvere e convertire le variabili di sostituzione solo dopo averne assegnato i valori. Prima di iniziare la conversione della tua fonte TeradataFastLoadscript di lavoro, assicuratevi di assegnare valori per tutte le variabili di sostituzione. Puoi usareAWS SCTper configurare le variabili di sostituzione negli script Teradata.

Per configurare le variabili di sostituzione nel tuoFastLoadscript di lavoro

1. Quando aggiungi la tua fonte TeradataFastLoadscript di lavoro per i tuoiAWS SCTprogetto, scegliVengono utilizzate variabili di sostituzione. Per ulteriori informazioni sull'aggiunta di questi script, vedere[AggiungendoFastLoadscript di lavoro per i tuoiAWS SCTprogetto](#).
2. PerDefinire il formato variabile, inserisci un'espressione regolare che corrisponda a tutte le variabili di sostituzione nello script.

Ad esempio, se i nomi delle variabili di sostituzione iniziano con\${e termina con}, usa il\$\{\w + \}espressione regolare. Per abbinare le variabili di sostituzione che iniziano con il simbolo del dollaro o con il simbolo di percentuale, usa\$\w + | \% \w +espressione regolare.

Le espressioni regolari in AWS SCT sono conformi alla sintassi delle espressioni regolari Java. Per ulteriori informazioni, vedere[Modello di classe java.util.regex](#)nella documentazione Java.

3. ScegliOKper caricare script sul tuoAWS SCTprogetto, quindi scegliOKper chiudere ilCaricare gli scriptfinestra.

4. Nel pannello di sinistra, espandi `Scriptnode`. Scegli `FastLoad`, quindi scegli la tua cartella con gli script. Apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli `Esporta variabili sotto Variabili di sostituzione`.

Inoltre, puoi esportare variabili di sostituzione per uno script. Espandi la cartella con gli script, scegli il tuo script, apri il menu contestuale (clic con il pulsante destro del mouse) e scegli `Esporta variabili sotto Variabili di sostituzione`.

5. Immettete il nome del file con valori separati da virgole (CSV) per salvare le variabili di sostituzione, quindi scegliete `Salva`.
6. Apri questo file CSV e inserisci i valori per le variabili di sostituzione.

A seconda del sistema operativo, AWS SCT utilizza diversi formati per il file CSV. I valori nel file possono essere racchiusi tra virgolette o meno. Assicurati di utilizzare lo stesso formato per i valori delle variabili di sostituzione degli altri valori nel file. AWS SCT non è possibile importare il file CSV con valori in formati diversi.

7. Salva il file CSV.
8. Nel pannello di sinistra, espandi `Scriptnode`. Scegli `FastLoad`, quindi scegli il tuo copione. Apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli `Importa variabili sotto Variabili di sostituzione`.
9. Scegli il tuo file CSV, quindi scegli `Aperto`.
10. Scegli `Variabili per visualizzare tutte le variabili di sostituzione scoperte e i relativi valori`.

Conversione di Teradata `FastLoad` script di lavoro con AWS SCT

Di seguito, scopri come convertire Teradata `FastLoad` lavoro su Amazon Redshift RSQL utilizzando AWS SCT.

Per convertire un Teradata `FastLoad` script di lavoro per Amazon Redshift RSQL

1. Aggiungi il tuo `FastLoad` script di lavoro per i tuoi AWS SCT progetto. Per ulteriori informazioni, consulta [Aggiungendo FastLoad script di lavoro per i tuoi AWS SCT progetto](#).
2. Configura le variabili di sostituzione. Per ulteriori informazioni, consulta [Configurazione delle variabili di sostituzione in FastLoad script di lavoro](#).
3. Nel pannello di sinistra, espandi `Scriptnode`.
4. Completa una delle seguenti operazioni:

- Per convertire un singolo FastLoad script, espandi FastLoad nodo, scegli lo script da convertire, quindi scegli Converti script dal menu contestuale (tasto destro).
- Per nascondere più script, assicurati di aver selezionato tutti gli script da convertire. Scegli FastLoad, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli Converti script. Effettuare quindi una delle seguenti operazioni:
 - Se archivi il tuo file di dati di origine su Amazon S3, scegli Percorso dell'oggetto S3 per Posizione del file di dati di origine.

Inserisci valori per Cartella bucket Amazon S3 e Bucket Amazon S3 per file manifest per il tuo file di dati di origine.

- Se non memorizzi il file dei dati di origine su Amazon S3, scegli Indirizzo dell'host per Posizione del file di dati di origine.

Inserisci valori per URL o indirizzo IP dell'host, Nome di accesso utente host, e Bucket Amazon S3 per file manifest per il tuo file di dati di origine.

5. Scegli OK.

AWS SCT converte tutti i Teradata selezionati FastLoad script di lavoro in un formato compatibile con Amazon Redshift RSQL. Trova i tuoi script convertiti nel Script nodo nel pannello del database di destinazione.

- ## 6. Modifica gli script RSQL di Amazon Redshift convertiti o salvati. Per ulteriori informazioni, consulta [Modifica e salvataggio dei file convertiti FastLoad script di lavoro](#).

Gestione di Teradata FastLoad script di lavoro con AWS SCT

Puoi aggiungere più Teradata FastLoad script di lavoro o rimuovi un FastLoad script di lavoro dal tuo AWS SCT progetto.

Per aggiungere un nuovo FastLoad script di lavoro per il tuo AWS SCT progetto

1. Espandi Script nodo nel pannello di sinistra.
2. Scegli la FastLoad nodo e apri il menu contestuale (clic con il pulsante destro del mouse).
3. Scegli Caricare gli script.
4. Inserisci le informazioni necessarie per aggiungerne una nuova FastLoad script di lavoro e configurazione delle variabili di sostituzione. Per ulteriori informazioni, consultare

[AggiungendoFastLoadscript di lavoro per i tuoiAWS SCTprogetto](#) e [Configurazione delle variabili di sostituzione inFastLoadscript di lavoro](#).

Per rimuovere unFastLoadscript di lavoro dal tuoAWS SCTprogetto

1. EspandiFastLoadnodo sottoScriptnel pannello di sinistra.
2. Scegli lo script da rimuovere e apri il menu contestuale (clic con il pulsante destro del mouse).
3. ScegliElimina script.

Creazione di un rapporto di valutazione per un TeradataFastLoadconversione di script di lavoro conAWS SCT

LaFastLoadrapporto di valutazione della conversione degli script di lavoro fornisce informazioni sulla conversione diFastLoadcomandi e istruzioni SQL. La conversione avviene dagli script di origine in un formato compatibile con Amazon Redshift RSQL. Il rapporto di valutazione include elementi di azione perFastLoadcomandi e istruzioni SQL cheAWS SCTimpossibile convertire.

Per creare un rapporto di valutazione della conversione degli script per un TeradataFastLoadoccupazione

1. EspandiFastLoadnodo sottoScriptnel pannello di sinistra.
2. Scegli lo script da convertire, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegliCrea rapporto.
3. Visualizza laRiepilogolinguetta.

LaRiepilogola scheda mostra le informazioni di riepilogo delFastLoadrapporto di valutazione del job script. Include risultati di conversione per tuttiFastLoadcomandi e istruzioni SQL dai tuoi script di origine.

4. (Facoltativo) Salvare una copia locale diFastLoadrapporto di valutazione della conversione degli script di lavoro come file PDF o file con valori separati da virgole (CSV):
 - Per salvare ilFastLoadrapporto di valutazione della conversione degli script di lavoro come file PDF, scegliSalva in PDFin alto a destra.

Il file PDF contiene il riepilogo esecutivo, le azioni e i consigli per la conversione degli script.

- Per salvare il rapporto di valutazione della conversione degli script di lavoro come file CSV, scegli **Salva in formato CSV** in alto a destra.

Il file CSV contiene le azioni, le azioni consigliate e una stima della complessità degli sforzi manuali necessari per convertire gli script.

5. Scegli la **Oggetti d'azione** linguetta. Questa scheda contiene un elenco di articoli che richiedono la conversione manuale in Amazon Redshift RSQL. Quando si seleziona un'azione dall'elenco, AWS SCT evidenzia l'articolo dalla tua fonte script di lavoro a cui si applica l'azione.

Modifica e salvataggio dei tuoi Teradata convertiti FastLoad script di lavoro con AWS SCT

Puoi modificare gli script convertiti nel pannello inferiore del tuo AWS SCT progetto. AWS SCT memorizza lo script modificato come parte del progetto.

Per salvare gli script convertiti

1. Espandi **Script RSQL** nodo sotto **Script** nel pannello del database di destinazione.
2. Scegli lo script convertito, apri il menu contestuale (clic con il pulsante destro del mouse) e scegli **Salva script**.
3. Inserisci il percorso della cartella per salvare lo script convertito e scegli **Salva**.

AWS SCT salva lo script convertito in un file e apre questo file.

Conversione di Teradata MultiLoad script di lavoro su Amazon Redshift RSQL con AWS SCT

Puoi usare AWS SCT per convertire Teradata MultiLoad script di lavoro per Amazon Redshift RSQL.

Un Teradata MultiLoad script di lavoro è un set di comandi per la manutenzione in batch del database Teradata. Un Teradata MultiLoad l'operazione di importazione esegue una serie di diverse operazioni di inserimento, aggiornamento ed eliminazione su un massimo di cinque tabelle e visualizzazioni diverse. Teradata MultiLoad le attività di eliminazione possono rimuovere un numero elevato di righe da una singola tabella.

AWS SCT converte Teradata MultiLoad comandi e istruzioni SQL in un formato compatibile con Amazon Redshift RSQL. Dopo aver migrato il database Teradata su Amazon Redshift, utilizza questi script convertiti per gestire i dati nel tuo database Amazon Redshift.

Argomenti

- [Aggiungendo MultiLoad script di lavoro per i tuoi AWS SCT progetto](#)
- [Configurazione delle variabili di sostituzione in Teradata MultiLoad script di lavoro con AWS SCT](#)
- [Conversione di Teradata MultiLoad script di lavoro con AWS SCT](#)
- [Gestione di Teradata MultiLoad script di lavoro con AWS SCT](#)
- [Creazione di un rapporto di valutazione per un Teradata MultiLoad conversione di script di lavoro con AWS SCT](#)
- [Modifica e salvataggio dei tuoi Teradata convertiti MultiLoad script di lavoro con AWS SCT](#)

Aggiungendo MultiLoad script di lavoro per i tuoi AWS SCT progetto

È possibile aggiungere più script a uno solo AWS SCT progetto.

Per aggiungere un MultiLoad script di lavoro per il tuo AWS SCT progetto

1. Crea un nuovo progetto in AWS SCT o apri un progetto esistente. Per ulteriori informazioni, consulta [the section called "Creazione di un progetto"](#).
2. Scegli **Aggiungi fonte** dal menu, quindi scegli **Teradata** per aggiungere il tuo database di origine al progetto. Per ulteriori informazioni, consulta [Utilizzo di Teradata come fonte](#).
3. Scegli **Aggiungi obiettivo** dal menu e per aggiungere un database Amazon Redshift di destinazione al tuo AWS SCT progetto.

Puoi utilizzare una piattaforma di database di destinazione Amazon Redshift virtuale. Per ulteriori informazioni, consulta [Utilizzo di obiettivi virtuali](#).

4. Crea una nuova regola di mappatura che includa il tuo database Teradata di origine e il tuo target Amazon Redshift. Per ulteriori informazioni, consulta [Aggiungere una nuova regola di mappatura](#).
5. Sul **Visualizza** menu, scegli **Vista principale**.
6. Nel pannello di sinistra, espandi **Script nodo**.
7. Scegli **MultiLoad**, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli **Caricare gli script**.

8. Inserisci la posizione della tua fonte TeradataMultiLoadscript di lavoro e scegli **Seleziona cartella**.

AWS SCT visualizza il **Caricare gli script** finestra.
9. Completa una delle seguenti operazioni:
 - Se il tuo TeradataMultiLoad gli script di lavoro non includono le variabili di sostituzione, scegli **Nessuna variabile di sostituzione**, quindi scegli **OK** per aggiungere script al tuo AWS SCT progetto.
 - Se il tuo TeradataMultiLoad gli script di lavoro includono le variabili di sostituzione, configurano le variabili di sostituzione. Per ulteriori informazioni, consulta [Configurazione delle variabili di sostituzione in MultiLoadscript di lavoro](#).

Configurazione delle variabili di sostituzione in TeradataMultiLoadscript di lavoro con AWS SCT

Il tuo TeradataMultiLoad gli script di lavoro potrebbero includere variabili di sostituzione. Ad esempio, è possibile utilizzare un singolo script con variabili di sostituzione per caricare dati su database diversi.

Prima di eseguire un MultiLoad job script con variabili di sostituzione, assicurati di assegnare i valori per tutte le variabili. Per fare ciò, puoi usare altri strumenti o applicazioni come uno script Bash, UC4 (Automatic) e così via.

AWS SCT può risolvere e convertire le variabili di sostituzione solo dopo averne assegnato i valori. Prima di iniziare la conversione della tua fonte TeradataMultiLoadscript di lavoro, assicurati di aver assegnato valori per tutte le variabili di sostituzione. Puoi usare AWS SCT per configurare le variabili di sostituzione negli script Teradata.

Per configurare le variabili di sostituzione nel tuo MultiLoadscript di lavoro

1. Quando aggiungi la tua fonte TeradataMultiLoadscript di lavoro per i tuoi AWS SCT progetto, scegli **Vengono utilizzate variabili di sostituzione**. Per ulteriori informazioni sull'aggiunta di questi script, vedere [Aggiungendo MultiLoadscript di lavoro per i tuoi AWS SCT progetto](#).
2. Per **Definire il formato variabile**, inserisci un'espressione regolare che corrisponda a tutte le variabili di sostituzione nello script.

Ad esempio, se i nomi delle variabili di sostituzione iniziano con `{` e termina con `}`, usa il `\${\w+\}` espressione regolare. Per abbinare le variabili di sostituzione che iniziano con il simbolo del dollaro o con il simbolo di percentuale, usa `\$\w+|\%\w+` espressione regolare.

Le espressioni regolari in AWS SCT sono conformi alla sintassi delle espressioni regolari Java. Per ulteriori informazioni, vedere [Modello di classe java.util.regex](#) nella documentazione Java.

3. Scegli **OK** per caricare script sul tuo AWS SCT progetto, quindi scegli **OK** per chiudere il Caricare gli script finestra.
4. Scegli **Variabili** per visualizzare tutte le variabili di sostituzione scoperte e i relativi valori.
5. Per **Valore**, inserite il valore per la variabile di sostituzione.

Conversione di Teradata MultiLoad script di lavoro con AWS SCT

Di seguito, scopri come convertire Teradata MultiLoad lavoro su Amazon Redshift RSQL utilizzando AWS SCT.

Per convertire un Teradata MultiLoad script di lavoro per Amazon Redshift RSQL

1. Aggiungi il tuo MultiLoad script di lavoro per i tuoi AWS SCT progetto. Per ulteriori informazioni, consulta [Aggiungendo MultiLoad script di lavoro per i tuoi AWS SCT progetto](#).
2. Configura le variabili di sostituzione e inserisci i loro valori. Per ulteriori informazioni, consulta [Configurazione delle variabili di sostituzione in MultiLoad script di lavoro](#).
3. Nel pannello di sinistra, espandi **Script nodo**.
4. Completa una delle seguenti operazioni:
 - Per convertire un singolo MultiLoad job script, espandi **MultiLoad nodo**, scegli lo script da convertire, quindi scegli **Converti script** dal menu contestuale (tasto destro).
 - Per nascondere più script, assicurati di aver selezionato tutti gli script da convertire. Scegli **MultiLoad**, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegli **Converti script**.
5. Completa una delle seguenti operazioni:
 - Se archivi il tuo file di dati di origine su Amazon S3, scegli **Percorso dell'oggetto S3** per **Posizione** del file di dati di origine.

EntraCartella bucket Amazon S3eBucket Amazon S3 per file manifestper il tuo file di dati di origine.

- Se non memorizzi il file dei dati di origine su Amazon S3, scegliIndirizzo dell'hostperPosizione del file di dati di origine.

EntraURL o indirizzo IP dell'host,Nome di accesso utente host, eBucket Amazon S3 per file manifestper il tuo file di dati di origine.

6. Scegli OK.

AWS SCTconverte tutti i Teradata selezionatiMultiLoadscript di lavoro in un formato compatibile con Amazon Redshift RSQL. Trova i tuoi script convertiti nelScriptnodo nel pannello del database di destinazione.

7. Modifica gli script RSQL di Amazon Redshift convertiti o salvati. Per ulteriori informazioni, consulta [Modifica e salvataggio dei file convertitiMultiLoadscript di lavoro](#).

Gestione di TeradataMultiLoadscript di lavoro conAWS SCT

Puoi aggiungere più TeradataMultiLoadscript di lavoro o rimuovi unMultiLoadscript di lavoro dal tuoAWS SCTprogetto.

Per aggiungere un nuovoMultiLoadscript di lavoro per il tuoAWS SCTprogetto

1. EspandiScriptnodo nel pannello di sinistra.
2. Scegli laMultiLoadnodo e apri il menu contestuale (clic con il pulsante destro del mouse).
3. ScegliCaricare gli script.
4. Inserisci le informazioni necessarie per aggiungerne una nuovaMultiLoadscript di lavoro e configurazione delle variabili di sostituzione. Per ulteriori informazioni, consultare [AggiungendoMultiLoadscript di lavoro per i tuoiAWS SCTprogetto](#) e [Configurazione delle variabili di sostituzione inMultiLoadscript di lavoro](#).

Per rimuovere unMultiLoadscript di lavoro dal tuoAWS SCTprogetto

1. EspandiMultiLoadnodo sottoScriptnel pannello di sinistra.
2. Scegli lo script da rimuovere e apri il menu contestuale (clic con il pulsante destro del mouse).
3. ScegliElimina script.

Creazione di un rapporto di valutazione per un TeradataMultiLoadconversione di script di lavoro conAWS SCT

LaMultiLoadrapporto di valutazione della conversione degli script di lavoro fornisce informazioni sulla conversione diMultiLoadcomandi e istruzioni SQL. La conversione avviene dagli script di origine ai comandi RSQL di Amazon Redshift e alle istruzioni SQL per Amazon Redshift. Il rapporto di valutazione include elementi di azione perMultiLoadcomandi e istruzioni SQL cheAWS SCTimpossibile convertire.

Per creare un rapporto di valutazione della conversione degli script per un TeradataMultiLoadoccupazione

1. EspandiMultiLoadnodo sottoScriptnel pannello di sinistra.
2. Scegli gli script per cui creare il rapporto di valutazione, apri il menu contestuale (clic con il pulsante destro del mouse), quindi scegliCrea rapporto.
3. Visualizza laRiepilogolinguetta. LaRiepilogola scheda mostra le informazioni di riepilogo delMultiLoadrapporto di valutazione del job script. Include risultati di conversione per tuttiMultiLoadcomandi e istruzioni SQL dai tuoi script di origine.
4. (Facoltativo) Salvare una copia locale diMultiLoadrapporto di valutazione della conversione degli script di lavoro come file PDF o file con valori separati da virgole (CSV):
 - Per salvare ilMultiLoadrapporto di valutazione della conversione degli script di lavoro come file PDF, scegliSalva in PDFin alto a destra.

Il file PDF contiene il riepilogo esecutivo, le azioni e i consigli per la conversione degli script.
 - Per salvare ilMultiLoadrapporto di valutazione della conversione degli script di lavoro in file CSV, scegliSalva in formato CSVin alto a destra.

AWS SCTcrea due file CSV. Questi file contengono il riepilogo esecutivo, le azioni, le azioni consigliate e una stima della complessità degli sforzi manuali necessari per convertire gli script.
5. Scegli laOggetti d'azionelinguetta. Questa scheda contiene un elenco di articoli che richiedono la conversione manuale in Amazon Redshift RSQL. Quando si seleziona un'azione dall'elenco,AWS SCTevidenzia l'articolo dalla tua fonteMultiLoadscript di lavoro a cui si applica l'azione.

Modifica e salvataggio dei tuoi Teradata convertiti MultiLoadscript di lavoro con AWS SCT

Puoi modificare gli script convertiti nel pannello inferiore del tuo AWS SCT progetto. AWS SCT memorizza lo script modificato come parte del progetto.

Per salvare gli script convertiti

1. Espandi Script RSQL nodo sotto Script nel pannello del database di destinazione.
2. Scegli lo script convertito, apri il menu contestuale (clic con il pulsante destro del mouse) e scegli Salva script.
3. Inserisci il percorso della cartella per salvare lo script convertito e scegli Salva.

AWS SCT salva lo script convertito in un file e apre questo file.

Migrazione di framework di big data con AWS Schema Conversion Tool

Puoi usare AWS Schema Conversion Tool (AWS SCT) per migrare i framework di big data su. Cloud AWS

Attualmente, AWS SCT supporta la migrazione di cluster Hadoop su Amazon EMR e Amazon S3. Questo processo di migrazione include i servizi Hive e HDFS.

Inoltre, puoi utilizzarlo AWS SCT per automatizzare la conversione dei tuoi flussi di lavoro di orchestrazione Apache Oozie in. AWS Step Functions

Argomenti

- [Migrazione di Apache Hadoop su Amazon EMR con AWS Schema Conversion Tool](#)
- [Convertire Apache Oozie in con AWS Step FunctionsAWS Schema Conversion Tool](#)

Migrazione di Apache Hadoop su Amazon EMR con AWS Schema Conversion Tool

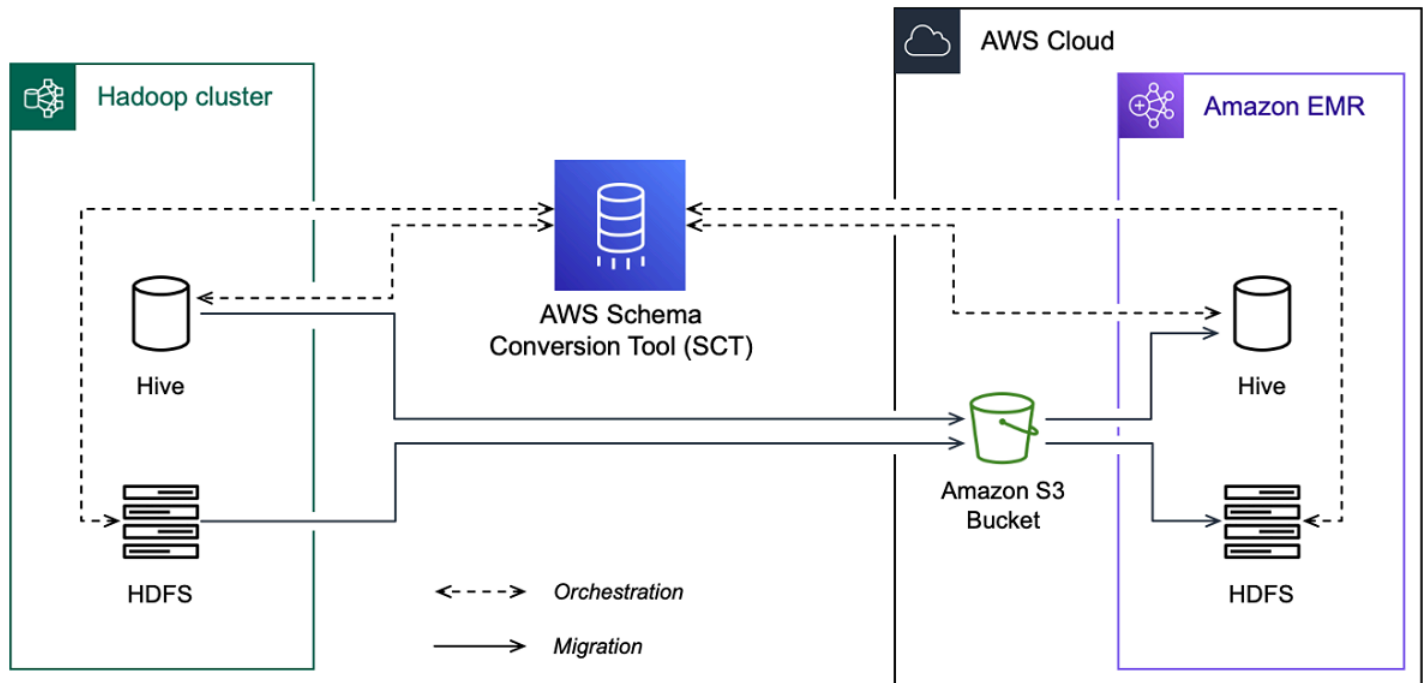
Per migrare i cluster Apache Hadoop, assicurati di utilizzare AWS SCT la versione 1.0.670 o successiva. Inoltre, acquisisci familiarità con l'interfaccia a riga di comando (CLI) di. AWS SCT Per ulteriori informazioni, consulta [AWS SCT Riferimento CLI](#).

Argomenti

- [Panoramica della migrazione](#)
- [Passaggio 1: connessi ai tuoi cluster Hadoop](#)
- [Fase 2: Impostazione delle regole di mappatura](#)
- [Fase 3: Creazione di un rapporto di valutazione](#)
- [Fase 4: Esegui la migrazione del tuo cluster Apache Hadoop su Amazon EMR con AWS SCT](#)
- [Esecuzione dello script CLI](#)
- [Gestire il progetto di migrazione dei big data](#)

Panoramica della migrazione

L'immagine seguente mostra il diagramma dell'architettura della migrazione da Apache Hadoop ad Amazon EMR.



AWS SCT esegue la migrazione di dati e metadati dal cluster Hadoop di origine a un bucket Amazon S3. Successivamente, AWS SCT utilizza i metadati Hive di origine per creare oggetti di database nel servizio Amazon EMR Hive di destinazione. Facoltativamente, puoi configurare Hive per utilizzare AWS Glue Data Catalog come metastore. In questo caso, AWS SCT migra i metadati Hive di origine su AWS Glue Data Catalog.

Quindi, puoi utilizzare AWS SCT per migrare i dati da un bucket Amazon S3 al servizio Amazon EMR HDFS di destinazione. In alternativa, puoi lasciare i dati nel tuo bucket Amazon S3 e utilizzarlo come archivio di dati per i tuoi carichi di lavoro Hadoop.

Per avviare la migrazione Hadoop, crei ed esegui il tuo script AWS SCT CLI. Questo script include il set completo di comandi per eseguire la migrazione. Puoi scaricare e modificare un modello dello script di migrazione Hadoop. Per ulteriori informazioni, consulta [Ottenere scenari CLI](#).

Assicurati che lo script includa i seguenti passaggi in modo da poter eseguire la migrazione da Apache Hadoop ad Amazon S3 e Amazon EMR.

Passaggio 1: connettiti ai tuoi cluster Hadoop

Per avviare la migrazione del tuo cluster Apache Hadoop, crea un nuovo progetto. AWS SCT Quindi, connettiti ai tuoi cluster di origine e destinazione. Assicurati di creare e fornire AWS le risorse di destinazione prima di iniziare la migrazione.

In questo passaggio, si utilizzano i seguenti comandi AWS SCT CLI.

- `CreateProject`— per creare un nuovo AWS SCT progetto.
- `AddSourceCluster`— per connettersi al cluster Hadoop di origine del AWS SCT progetto.
- `AddSourceClusterHive`— per connettersi al servizio Hive di origine del progetto.
- `AddSourceClusterHDFS`— per connettersi al servizio HDFS di origine del progetto.
- `AddTargetCluster`— per connettersi al cluster Amazon EMR di destinazione del progetto.
- `AddTargetClusterS3`— per aggiungere il bucket Amazon S3 al tuo progetto.
- `AddTargetClusterHive`— per connettersi al servizio Hive di destinazione del progetto
- `AddTargetClusterHDFS`— per connettersi al servizio HDFS di destinazione del progetto

Per esempi di utilizzo di questi comandi AWS SCT CLI, vedere [Usare Apache Hadoop come fonte](#).

Quando esegui il comando che si connette a un cluster di origine o di destinazione, AWS SCT tenta di stabilire la connessione a questo cluster. Se il tentativo di connessione fallisce, AWS SCT interrompe l'esecuzione dei comandi dallo script CLI e visualizza un messaggio di errore.

Fase 2: Impostazione delle regole di mappatura

Dopo esserti connesso ai cluster di origine e di destinazione, configura le regole di mappatura. Una regola di mappatura definisce l'obiettivo di migrazione per un cluster di origine. Assicurati di aver impostato le regole di mappatura per tutti i cluster di origine che hai aggiunto al tuo AWS SCT progetto. Per ulteriori informazioni sulle regole di mappatura, vedere [Creazione di regole di mappatura in AWS SCT](#).

In questo passaggio, si utilizza il `AddServerMapping` comando. Questo comando utilizza due parametri che definiscono i cluster di origine e di destinazione. È possibile utilizzare il `AddServerMapping` comando con il percorso esplicito degli oggetti del database o con i nomi di un oggetto. Per la prima opzione, includi il tipo di oggetto e il suo nome. Per la seconda opzione, includi solo i nomi degli oggetti.

- `sourceTreePath`— il percorso esplicito degli oggetti del database di origine.
`targetTreePath`— il percorso esplicito degli oggetti del database di destinazione.
- `sourceNamePath`— il percorso che include solo i nomi degli oggetti di origine.
`targetNamePath`— il percorso che include solo i nomi degli oggetti bersaglio.

L'esempio di codice seguente crea una regola di mappatura utilizzando percorsi espliciti per il database `testdb` Hive di origine e il cluster EMR di destinazione.

```
AddServerMapping
  -sourceTreePath: 'Clusters.HADOOP_SOURCE.HIVE_SOURCE.Databases.testdb'
  -targetTreePath: 'Clusters.HADOOP_TARGET.HIVE_TARGET'
/
```

È possibile utilizzare questo esempio e i seguenti in Windows. Per eseguire i comandi CLI in Linux, assicurati di aver aggiornato i percorsi dei file in modo appropriato per il tuo sistema operativo.

L'esempio di codice seguente crea una regola di mappatura utilizzando i percorsi che includono solo i nomi degli oggetti.

```
AddServerMapping
  -sourceNamePath: 'HADOOP_SOURCE.HIVE_SOURCE.testdb'
  -targetNamePath: 'HADOOP_TARGET.HIVE_TARGET'
/
```

Puoi scegliere Amazon EMR o Amazon S3 come destinazione per il tuo oggetto di origine. Per ogni oggetto di origine, puoi scegliere un solo obiettivo in un singolo AWS SCT progetto. Per modificare l'obiettivo di migrazione per un oggetto di origine, elimina la regola di mappatura esistente e crea una nuova regola di mappatura. Per eliminare una regola di mappatura, utilizzare il `DeleteServerMapping` comando. Questo comando utilizza uno dei due parametri seguenti.

- `sourceTreePath`— il percorso esplicito degli oggetti del database di origine.
- `sourceNamePath`— il percorso che include solo i nomi degli oggetti di origine.

Per ulteriori informazioni sui `DeleteServerMapping` comandi `AddServerMapping` and, vedete la [AWS Schema Conversion Tool CLI Reference](#).

Fase 3: Creazione di un rapporto di valutazione

Prima di iniziare la migrazione, ti consigliamo di creare un rapporto di valutazione. Questo rapporto riassume tutte le attività di migrazione e descrive in dettaglio le azioni che emergeranno durante la migrazione. Per assicurarti che la migrazione non vada a buon fine, visualizza questo rapporto e affronta le azioni da intraprendere prima della migrazione. Per ulteriori informazioni, consulta [Rapporti di valutazione della migrazione](#).

In questo passaggio, si utilizza il `CreateMigrationReport` comando. Questo comando utilizza due parametri. Il `treePath` parametro è obbligatorio e il `forceMigrate` parametro è facoltativo.

- `treePath`— il percorso esplicito degli oggetti del database di origine per i quali si salva una copia del rapporto di valutazione.
- `forceMigrate`— se impostato su `true`, AWS SCT continua la migrazione anche se il progetto include una cartella HDFS e una tabella Hive che fanno riferimento allo stesso oggetto. Il valore di default è `false`.

È quindi possibile salvare una copia del rapporto di valutazione come file PDF o con valori separati da virgole (CSV). A tale scopo, utilizzare il `SaveReportCSV` comando `SaveReportPDF` or.

Il `SaveReportPDF` comando salva una copia del rapporto di valutazione come file PDF. Questo comando utilizza quattro parametri. Il `file` parametro è obbligatorio, gli altri parametri sono opzionali.

- `file`— il percorso del file PDF e il suo nome.
- `filter`— il nome del filtro che hai creato in precedenza per definire l'ambito degli oggetti di origine da migrare.
- `treePath`— il percorso esplicito degli oggetti del database di origine per i quali si salva una copia del rapporto di valutazione.
- `namePath`— il percorso che include solo i nomi degli oggetti target per i quali si salva una copia del rapporto di valutazione.

Il `SaveReportCSV` comando salva il rapporto di valutazione in tre file CSV. Questo comando utilizza quattro parametri. Il `directory` parametro è obbligatorio, gli altri parametri sono opzionali.

- `directory`— il percorso della cartella in cui vengono AWS SCT salvati i file CSV.

- `filter`— il nome del filtro che hai creato in precedenza per definire l'ambito degli oggetti di origine da migrare.
- `treePath`— il percorso esplicito degli oggetti del database di origine per i quali si salva una copia del rapporto di valutazione.
- `namePath`— il percorso che include solo i nomi degli oggetti target per i quali si salva una copia del rapporto di valutazione.

L'esempio di codice seguente salva una copia del rapporto di valutazione nel `c:\sct\ar.pdf` file.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

L'esempio di codice seguente salva una copia del rapporto di valutazione come file CSV nella `c:\sct` cartella.

```
SaveReportCSV
-file:'c:\sct'
/
```

Per ulteriori informazioni sui `SaveReportCSV` comandi `SaveReportPDF` and, vedete la [AWS Schema Conversion Tool CLI Reference](#).

Fase 4: Esegui la migrazione del tuo cluster Apache Hadoop su Amazon EMR con AWS SCT

Dopo aver configurato il AWS SCT progetto, avvia la migrazione del tuo cluster Apache Hadoop locale su. Cloud AWS

In questo passaggio, si utilizzano i `ResumeMigration` comandi `MigrateMigrationStatus`, e.

Il `Migrate` comando esegue la migrazione degli oggetti di origine nel cluster di destinazione. Questo comando utilizza quattro parametri. Assicuratevi di specificare il `treePath` parametro `filter` or. Gli altri parametri sono opzionali.

- `filter`— il nome del filtro che hai creato in precedenza per definire l'ambito degli oggetti di origine da migrare.

- `treePath`— il percorso esplicito degli oggetti del database di origine per i quali si salva una copia del rapporto di valutazione.
- `forceLoad`— se impostato su `true`, carica AWS SCT automaticamente gli alberi dei metadati del database durante la migrazione. Il valore di default è `false`.
- `forceMigrate`— se impostato su `true`, AWS SCT continua la migrazione anche se il progetto include una cartella HDFS e una tabella Hive che fanno riferimento allo stesso oggetto. Il valore di default è `false`.

Il `MigrationStatus` comando restituisce le informazioni sull'avanzamento della migrazione. Per eseguire questo comando, inserisci il nome del tuo progetto di migrazione per il `name` parametro. Hai specificato questo nome nel `CreateProject` comando.

Il `ResumeMigration` comando riprende la migrazione interrotta che hai avviato utilizzando il comando `Migrate`. Il `ResumeMigration` comando non utilizza parametri. Per riprendere la migrazione, è necessario connettersi ai cluster di origine e di destinazione. Per ulteriori informazioni, consulta [Gestire il progetto di migrazione](#).

Il seguente esempio di codice esegue la migrazione dei dati dal servizio HDFS di origine ad Amazon EMR.

```
Migrate
-treePath: 'Clusters.HADOOP_SOURCE.HDFS_SOURCE'
-forceMigrate: 'true'
/
```

Esecuzione dello script CLI

Dopo aver modificato lo script AWS SCT CLI, salvalo come file con l' `.scts` estensione. Ora puoi eseguire lo script dalla app cartella del tuo percorso di AWS SCT installazione. A tale scopo, utilizzare il comando seguente.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\hadoop.scts"
```

Nell'esempio precedente, sostituisci *script_path con il percorso* del tuo file con lo script CLI. Per ulteriori informazioni sull'esecuzione degli script CLI in AWS SCT, vedere. [Modalità script](#)

Gestire il progetto di migrazione dei big data

Dopo aver completato la migrazione, puoi salvare e modificare il tuo AWS SCT progetto per un uso futuro.

Per salvare il tuo AWS SCT progetto, usa il `SaveProject` comando. Questo comando non utilizza parametri.

Il seguente esempio di codice salva il AWS SCT progetto.

```
SaveProject  
/
```

Per aprire il tuo AWS SCT progetto, usa il `OpenProject` comando. Questo comando utilizza un parametro obbligatorio. Come `file` parametro, inserisci il percorso del file di AWS SCT progetto e il suo nome. Hai specificato il nome del progetto nel `CreateProject` comando. Assicurati di aggiungere l'.`sct` estensione al nome del file di progetto per eseguire il `OpenProject` comando.

L'esempio di codice seguente apre il `hadoop_emr` progetto dalla `c:\sct` cartella.

```
OpenProject  
-file: 'c:\sct\hadoop_emr.scts'  
/
```

Dopo aver aperto il AWS SCT progetto, non è necessario aggiungere i cluster di origine e di destinazione perché li hai già aggiunti al progetto. Per iniziare a lavorare con i tuoi cluster di origine e destinazione, devi connetterti a loro. A tale scopo, si utilizzano i `ConnectTargetCluster` comandi `ConnectSourceCluster` and. Questi comandi utilizzano gli stessi parametri dei `AddTargetCluster` comandi `AddSourceCluster` and. Puoi modificare il tuo script CLI e sostituire il nome di questi comandi lasciando l'elenco dei parametri senza modifiche.

Il seguente esempio di codice si connette al cluster Hadoop di origine.

```
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'  
/  

```

Il seguente esempio di codice si connette al cluster Amazon EMR di destinazione.

```
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'  
/  

```

Nell'esempio precedente, sostituisci *hadoop_address* con l'*indirizzo* IP del tuo cluster Hadoop. Se necessario, configura il valore della variabile porta. Quindi, sostituisci *hadoop_user* e *hadoop_password* con il nome del tuo utente Hadoop e la password per questo utente. In *path\ name*, inserisci il nome e il percorso del file PEM per il cluster Hadoop di origine. Per ulteriori informazioni sull'aggiunta dei cluster di origine e destinazione, consulta [Utilizzo di Apache Hadoop come fonte per AWS SCT](#).

Dopo aver effettuato la connessione ai cluster Hadoop di origine e di destinazione, è necessario connettersi ai servizi Hive e HDFS, nonché al bucket Amazon S3. A tale scopo, si utilizzano i ConnectTargetClusterS3 comandi ConnectSourceClusterHive ConnectSourceClusterHdfsConnectTargetClusterHive,ConnectTargetClusterHdfs,, e. Questi comandi utilizzano gli stessi parametri dei comandi che hai usato per aggiungere i servizi Hive e HDFS e il bucket Amazon S3 al tuo progetto. Modifica lo script CLI per sostituire il Add prefisso con Connect nei nomi dei comandi.

Convertire Apache Oozie in con AWS Step FunctionsAWS Schema Conversion Tool

Per convertire i flussi di lavoro di Apache Oozie, assicurati di utilizzare la AWS SCT versione 1.0.671 o superiore. Inoltre, acquisisci familiarità con l'interfaccia a riga di comando (CLI) di AWS SCT. Per ulteriori informazioni, consulta [AWS SCT Riferimento CLI](#).

Argomenti

- [Panoramica delle conversioni](#)
- [Passaggio 1: connettiti ai tuoi servizi di origine e destinazione](#)
- [Fase 2: Impostazione delle regole di mappatura](#)
- [Fase 3: Configurazione dei parametri](#)
- [Fase 4: Creazione di un rapporto di valutazione](#)
- [Passaggio 5: converti i tuoi flussi di lavoro di Apache Oozie in AWS Step FunctionsAWS SCT](#)
- [Esecuzione dello script CLI](#)
- [Nodi Apache Oozie che AWS SCT possono essere convertiti in AWS Step Functions](#)

Panoramica delle conversioni

Il codice sorgente di Apache Oozie include nodi di azione, nodi di flusso di controllo e proprietà dei job. I nodi di azione definiscono i processi che esegui nel tuo flusso di lavoro di Apache Oozie. Quando usi Apache Oozie per orchestrare il tuo cluster Apache Hadoop, un nodo di azione include un job Hadoop. I nodi del flusso di controllo forniscono un meccanismo per controllare il percorso del flusso di lavoro. I nodi del flusso di controllo includono nodi come `startend`, `decision`, `fork`, e `join`.

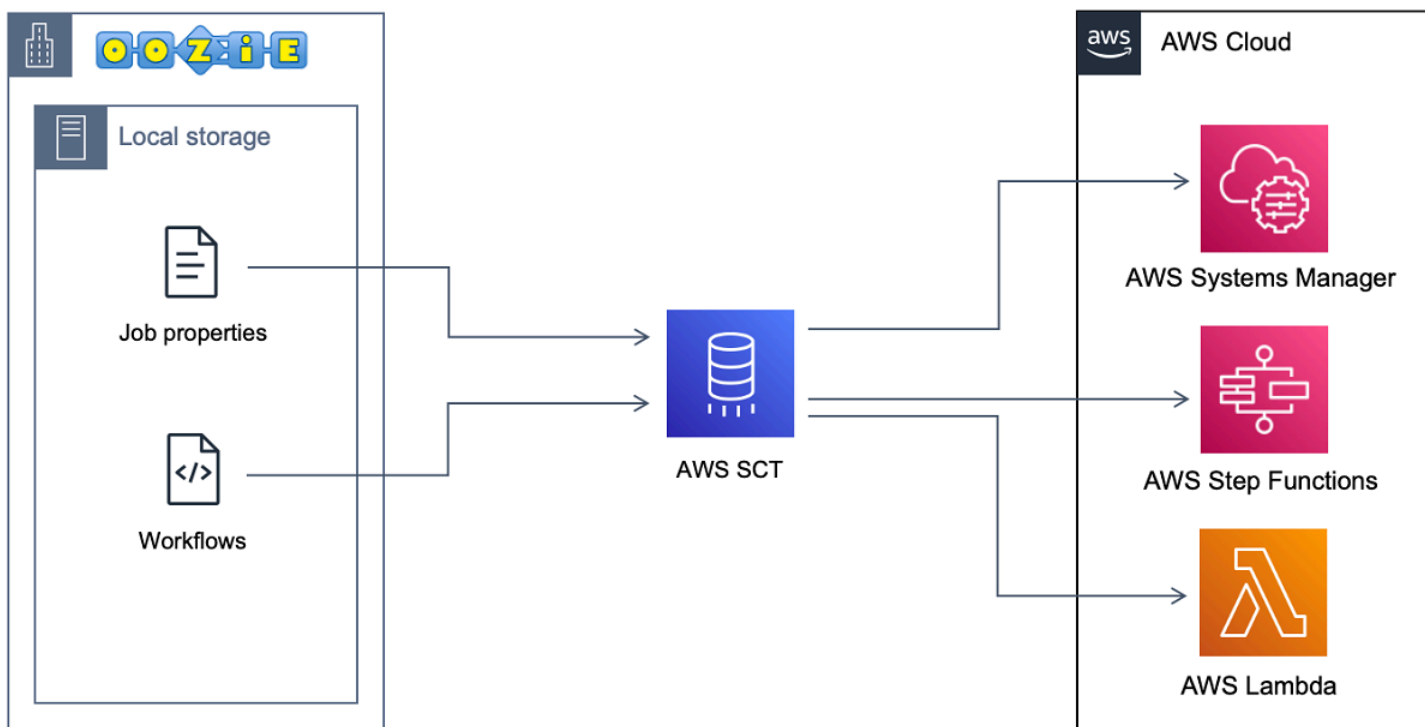
AWS SCT converte i nodi di azione di origine e i nodi del flusso di controllo in AWS Step Functions. In AWS Step Functions, definisci i tuoi flussi di lavoro in Amazon States Language (ASL). AWS SCT usa ASL per definire la tua macchina a stati, che è un insieme di stati, che può funzionare, determinare a quali stati passare successivamente, fermarsi con un errore e così via. Successivamente, AWS SCT carica i file JSON con le definizioni delle macchine a stati. Quindi, AWS SCT puoi usare il tuo ruolo AWS Identity and Access Management (IAM) per configurare le tue macchine a stato in AWS Step Functions. Per ulteriori informazioni, consulta [Cos'è AWS Step Functions?](#) nella Guida per gli sviluppatori AWS Step Functions.

Inoltre, AWS SCT crea un pacchetto di estensione con AWS Lambda funzioni che emulano le funzioni di origine che AWS Step Functions non sono supportate. Per ulteriori informazioni, consulta [Utilizzo dei pacchetti di AWS SCT estensione](#).

AWS SCT migra le proprietà lavorative di origine su AWS Systems Manager. Per memorizzare i nomi e i valori dei parametri, AWS SCT utilizza Parameter Store, una funzionalità di AWS Systems Manager. Per ulteriori informazioni, consulta [Che cos'è AWS Systems Manager?](#) nella Guida per l'utente di AWS Systems Manager.

Puoi usarlo AWS SCT per aggiornare automaticamente i valori e i nomi dei tuoi parametri. A causa delle differenze di architettura tra Apache Oozie e AWS Step Functions, potrebbe essere necessario configurare i parametri. AWS SCT può trovare un nome o un valore di parametro specificato nei file di origine e sostituirli con nuovi valori. Per ulteriori informazioni, consulta [Fase 3: Configurazione dei parametri](#).

L'immagine seguente mostra il diagramma dell'architettura della conversione di Apache Oozie in AWS Step Functions



Per avviare la conversione, crea ed esegui il tuo script AWS SCT CLI. Questo script include il set completo di comandi per eseguire la conversione. È possibile scaricare e modificare un modello dello script di conversione di Apache Oozie. Per ulteriori informazioni, consulta [Ottenere scenari CLI](#).

Assicurati che il tuo script includa i seguenti passaggi.

Passaggio 1: connettiti ai tuoi servizi di origine e destinazione

Per avviare la conversione del tuo cluster Apache Oozie, crea un nuovo AWS SCT progetto. Quindi, connettiti ai tuoi servizi di origine e destinazione. Assicurati di creare e fornire AWS le risorse di destinazione prima di iniziare la migrazione. Per ulteriori informazioni, consulta [Prerequisiti per l'utilizzo di Apache Oozie come sorgente](#).

In questo passaggio, si utilizzano i seguenti comandi AWS SCT CLI.

- `CreateProject`— per creare un nuovo AWS SCT progetto.
- `AddSource`— per aggiungere i file sorgente di Apache Oozie al tuo AWS SCT progetto.
- `ConnectSource`— per connettersi ad Apache Oozie come sorgente.
- `AddTarget`— da aggiungere AWS Step Functions come obiettivo di migrazione nel tuo progetto.
- `ConnectTarget`— a cui connettersi AWS Step Functions.

Per esempi di utilizzo di questi comandi AWS SCT CLI, vedere [Usare Apache Oozie come fonte](#).

Quando `ConnectTarget` esegui i comandi `ConnectSource` or, AWS SCT tenta di stabilire la connessione ai tuoi servizi. Se il tentativo di connessione fallisce, AWS SCT interrompe l'esecuzione dei comandi dallo script CLI e visualizza un messaggio di errore.

Fase 2: Impostazione delle regole di mappatura

Dopo esserti connesso ai servizi di origine e destinazione, configura le regole di mappatura. Una regola di mappatura definisce l'obiettivo di migrazione per i flussi di lavoro e i parametri di Apache Oozie di origine. Per ulteriori informazioni sulle regole di mappatura, vedere [Creazione di regole di mappatura in AWS SCT](#).

Per definire gli oggetti di origine e destinazione per la conversione, utilizzate il `AddServerMapping` comando. Questo comando utilizza due parametri: `sourceTreePath` e `targetTreePath`. I valori di questi parametri includono un percorso esplicito verso gli oggetti di origine e di destinazione. Affinché Apache Oozie possa essere AWS Step Functions convertito, questi parametri devono iniziare con `ETL`

L'esempio di codice seguente crea una regola di mappatura per `OOZIE` e `AWS_STEP_FUNCTIONS` oggetti. Hai aggiunto questi oggetti al tuo AWS SCT progetto utilizzando `AddTarget` i comandi `AddSource` e nel passaggio precedente.

```
AddServerMapping
  -sourceTreePath: 'ETL.APACHE_OOZIE'
  -targetTreePath: 'ETL.AWS_STEP_FUNCTIONS'
/
```

Per ulteriori informazioni sul `AddServerMapping` comando, consulta la [AWS Schema Conversion ToolCLI Reference](#).

Fase 3: Configurazione dei parametri

Se i flussi di lavoro di origine di Apache Oozie utilizzano parametri, potrebbe essere necessario modificarne i valori dopo la conversione in AWS Step Functions. Inoltre, potresti dover aggiungere nuovi parametri da utilizzare con il tuo AWS Step Functions.

In questo passaggio, si utilizzano i `AddTargetParameter` comandi `AddParameterMapping` and.

Per sostituire i valori dei parametri nei file sorgente, utilizzate il `AddParameterMapping` comando. AWS SCT analizza i file di origine, trova i parametri per nome o valore e ne modifica i valori. Puoi eseguire un singolo comando per scansionare tutti i tuoi file sorgente. L'ambito dei file da scansionare viene definito utilizzando uno dei primi tre parametri del seguente elenco. Questo comando utilizza fino a sei parametri.

- `filterName`— il nome del filtro per gli oggetti di origine. Puoi creare un filtro usando il `CreateFilter` comando.
- `treePath`— il percorso esplicito degli oggetti di origine.
- `namePath`— il percorso esplicito di un oggetto sorgente specifico.
- `sourceParameterName`— il nome del parametro di origine.
- `sourceValue`— il valore del parametro di origine.
- `targetValue`— il valore del parametro di destinazione.

L'esempio di codice seguente sostituisce tutti i parametri in cui il valore è uguale `c:\oozie\hive.py` al `s3://bucket-oozie/hive.py` valore.

```
AddParameterMapping
  -treePath: 'ETL.OOZIE.Applications'
  -sourceValue: 'c:\oozie\hive.py'
  -targetValue: 's3://bucket-oozie/hive.py'
```

/

L'esempio di codice seguente sostituisce tutti i parametri in cui il nome è uguale `nameNode` al `hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020` valore.

```
AddParameterMapping
  -treePath: 'ETL.00ZIE_SOURCE.Applications'
  -sourceParameter: 'nameNode'
  -targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'
```

/

L'esempio di codice seguente sostituisce tutti i parametri in cui il nome è uguale `nameNode` e il valore è uguale a `hdfs://ip-55.eu-west-1.compute.internal:8020` con il valore del `targetValue` parametro.

```
AddParameterMapping
  -treePath: 'ETL.00ZIE_SOURCE.Applications'
  -sourceParameter: 'nameNode'
  -sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'
  -targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'
```

/

Per aggiungere un nuovo parametro nei file di destinazione oltre a un parametro esistente dai file di origine, utilizzate il `AddTargetParameter` comando. Questo comando utilizza lo stesso set di parametri del `AddParameterMapping` comando.

L'esempio di codice seguente aggiunge il parametro `clusterId` target anziché il `nameNode` parametro.

```
AddTargetParameter
  -treePath: 'ETL.00ZIE_SOURCE.Applications'
  -sourceParameter: 'nameNode'
  -sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'
  -targetParameter: 'clusterId'
  -targetValue: '1234567890abcdef0'
```

/

Per ulteriori informazioni sui `CreateFilter` comandi `AddServerMapping`, `AddParameterMapping`, e `AddTargetParameter`, consulta la [AWS Schema Conversion Tool CLI Reference](#).

Fase 4: Creazione di un rapporto di valutazione

Prima di iniziare la conversione, ti consigliamo di creare un rapporto di valutazione. Questo rapporto riassume tutte le attività di migrazione e descrive in dettaglio le azioni che emergeranno durante la migrazione. Per assicurarti che la migrazione non vada a buon fine, visualizza questo rapporto e affronta le azioni da intraprendere prima della migrazione. Per ulteriori informazioni, consulta [Rapporti di valutazione della migrazione](#).

In questo passaggio, si utilizza il `CreateReport` comando. Questo comando utilizza due parametri. Il primo parametro descrive gli oggetti di origine per i quali AWS SCT crea un rapporto di valutazione. A tale scopo, utilizzare uno dei seguenti parametri: `filterName`, `treePath`, `onamePath`. Questo parametro è obbligatorio. Inoltre, puoi aggiungere un parametro booleano opzionale. `forceLoad` Se impostate questo parametro su `true`, carica AWS SCT automaticamente tutti gli oggetti secondari per l'oggetto di origine specificato nel `CreateReport` comando.

L'esempio di codice seguente crea un rapporto di valutazione per il `Applications` nodo dei file Oozie di origine.

```
CreateReport
  -treePath: 'ETL.APACHE_00ZIE.Applications'
/
```

È quindi possibile salvare una copia del rapporto di valutazione come file PDF o con valori separati da virgole (CSV). A tale scopo, utilizzare il `SaveReportCSV` comando `SaveReportPDF` or.

Il `SaveReportPDF` comando salva una copia del rapporto di valutazione come file PDF. Questo comando utilizza quattro parametri. Il `file` parametro è obbligatorio, gli altri parametri sono opzionali.

- `file`— il percorso del file PDF e il suo nome.
- `filter`— il nome del filtro che hai creato in precedenza per definire l'ambito degli oggetti di origine da migrare.
- `treePath`— il percorso esplicito degli oggetti del database di origine per i quali si salva una copia del rapporto di valutazione.
- `namePath`— il percorso che include solo i nomi degli oggetti target per i quali si salva una copia del rapporto di valutazione.

Il `SaveReportCSV` comando salva il rapporto di valutazione in file CSV. Questo comando utilizza quattro parametri. Il `directory` parametro è obbligatorio, gli altri parametri sono opzionali.

- `directory`— il percorso della cartella in cui vengono AWS SCT salvati i file CSV.
- `filter`— il nome del filtro che hai creato in precedenza per definire l'ambito degli oggetti di origine da migrare.
- `treePath`— il percorso esplicito degli oggetti del database di origine per i quali si salva una copia del rapporto di valutazione.
- `namePath`— il percorso che include solo i nomi degli oggetti target per i quali si salva una copia del rapporto di valutazione.

L'esempio di codice seguente salva una copia del rapporto di valutazione nel `c:\sct\ar.pdf` file.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

L'esempio di codice seguente salva una copia del rapporto di valutazione come file CSV nella `c:\sct` cartella.

```
SaveReportCSV
-file:'c:\sct'
/
```

Per ulteriori informazioni sui `SaveReportCSV` comandi `SaveReportPDF` e `CreateReport`, consulta la [AWS Schema Conversion Tool CLI Reference](#).

Passaggio 5: converti i tuoi flussi di lavoro di Apache Oozie in AWS Step FunctionsAWS SCT

Dopo aver configurato il AWS SCT progetto, converti il codice sorgente e applicalo a Cloud AWS.

In questo passaggio, si utilizzano i `ApplyToTarget` comandi `Convert` `SaveOnS3ConfigureStateMachine`, e.

Il `Migrate` comando esegue la migrazione degli oggetti di origine nel cluster di destinazione. Questo comando utilizza quattro parametri. Assicuratevi di specificare il `treePath` parametro `filter` or. Gli altri parametri sono opzionali.

- `filter`— il nome del filtro che hai creato in precedenza per definire l'ambito degli oggetti di origine da migrare.
- `namePath`— il percorso esplicito di un oggetto sorgente specifico.
- `treePath`— il percorso esplicito degli oggetti del database di origine per i quali si salva una copia del rapporto di valutazione.
- `forceLoad`— se impostato su `true`, carica AWS SCT automaticamente gli alberi dei metadati del database durante la migrazione. Il valore di default è `false`.

Il seguente esempio di codice converte i file dalla `Applications` cartella dei file Oozie di origine.

```
Convert
  -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

Quindi `SaveOnS3` carica le definizioni delle macchine a stato nel tuo bucket Amazon S3. Questo comando utilizza il `treePath` parametro. Per eseguire questo comando, usa la cartella di destinazione con le definizioni delle macchine a stati come valore di questo parametro.

Quanto segue carica la `State machine definitions` cartella dell'oggetto di `AWS_STEP_FUNCTIONS` destinazione nel bucket Amazon S3. AWS SCT utilizza il bucket Amazon S3 che hai archiviato nel profilo del AWS servizio durante la [Prerequisiti](#) fase.

```
SaveOnS3
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
/
```

Il `ConfigureStateMachine` comando configura le macchine a stati. Questo comando utilizza fino a sei parametri. Assicuratevi di definire l'ambito di destinazione utilizzando uno dei primi tre parametri del seguente elenco.

- `filterName`— il nome del filtro per gli oggetti bersaglio. Puoi creare un filtro usando il `CreateFilter` comando.
- `treePath`— il percorso esplicito verso gli oggetti bersaglio.
- `namePath`— il percorso esplicito verso uno specifico oggetto di destinazione.
- `iamRole`— l'Amazon Resource Name (ARN) del ruolo IAM che fornisce l'accesso alle tue step machine. Questo parametro è obbligatorio.

L'esempio di codice seguente configura le macchine a stati definite `AWS_STEP_FUNCTIONS` utilizzando il *ruolo IAM `role_name`*.

```
ConfigureStateMachine
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
  -role: 'arn:aws:iam::555555555555:role/role_name'
/
```

Il `ApplyToTarget` comando applica il codice convertito al server di destinazione. Per eseguire questo comando, utilizzate uno dei seguenti parametri: `filterName`, `treePath`, o `namePath` per definire gli oggetti di destinazione da applicare.

Il seguente esempio di codice applica la macchina a `app_wp` stati a `AWS Step Functions`.

```
ApplyToTarget
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machines.app_wp'
/
```

Per assicurarti che il codice convertito produca gli stessi risultati del codice sorgente, puoi usare il pacchetto di `AWS SCT` estensione. Questo è un insieme di `AWS Lambda` funzioni che emulano le funzioni di `Apache Oozie` che `AWS Step Functions` non sono supportate. Per installare questo pacchetto di estensione, puoi usare il `CreateLambdaExtPack` comando.

Questo comando utilizza fino a cinque parametri. Assicurati di utilizzare **`Oozie2SF`** per `extPackId`. In questo caso, `AWS SCT` crea un pacchetto di estensione per le funzioni sorgente di `Apache Oozie`.

- `extPackId`— l'identificatore univoco per un insieme di funzioni `Lambda`. Questo parametro è obbligatorio.
- `tempDirectory`— il percorso in cui è `AWS SCT` possibile archiviare i file temporanei. Questo parametro è obbligatorio.
- `awsProfile`— il nome del tuo `AWS` profilo.
- `lambdaExecRoles`— l'elenco di `Amazon Resource Names (ARN)` dei ruoli di esecuzione da utilizzare per le funzioni `Lambda`.
- `createInvokeRoleFlag`— la bandiera booleana che indica se creare un ruolo di esecuzione per `AWS Step Functions`.

Per installare e utilizzare il pacchetto di estensione, assicurati di fornire le autorizzazioni richieste. Per ulteriori informazioni, consulta [Autorizzazioni per l'utilizzo AWS Lambda delle funzioni nel pacchetto di estensione](#).

Per ulteriori informazioni sui CreateLambdaExtPack comandiConvert,, SaveOnS3ConfigureStateMachine, eApplyToTarget, consulta la [AWS Schema Conversion ToolCLI Reference](#).

Esecuzione dello script CLI

Dopo aver modificato lo script AWS SCT CLI, salvalo come file con l'.sctsestensione. Ora puoi eseguire lo script dalla app cartella del tuo percorso di AWS SCT installazione. A tale scopo, utilizzare il comando seguente.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\oozie.scts"
```

Nell'esempio precedente, sostituisci *script_path con il percorso* del tuo file con lo script CLI. Per ulteriori informazioni sull'esecuzione degli script CLI inAWS SCT, vedere. [Modalità script](#)

Nodi Apache Oozie che AWS SCT possono essere convertiti in AWS Step Functions

Puoi usarlo AWS SCT per convertire i nodi di azione e i nodi di flusso di controllo di Apache Oozie in. AWS Step Functions

I nodi di azione supportati includono quanto segue:

- Azione Hive
- Azione Hive2
- Accendi l'azione
- MapReduceAzione in streaming
- Azione Java
- Operazione DistCp
- Azione suina
- Azione Sqoop
- Azione FS

- **Azione Shell**

I nodi del flusso di controllo supportati includono quanto segue:

- Avvia azione
- Termina azione
- Azione di uccisione
- Azione decisionale
- Azione a forcella
- Unisciti all'azione

Uso di AWS SCT con AWS DMS

Utilizzo di unAWS SCT agente di replica conAWS DMS

Per migrazioni di database molto grandi, è possibile utilizzare unAWS SCT agente di replica (`aws-schema-conversion-tool-dms-agent`) per copiare i dati dal database locale nel database Amazon S3 oAWS SnowballDispositivo Edge. L'agente di replica funziona in combinazione con AWS DMS e può essere eseguito in background mentre AWS SCT è chiuso.

Quando lavori conAWS SnowballEdge,AWS SCT agente replica i dati nelAWS Snowballdispositivo. Il dispositivo viene quindi inviato aAWS e i dati vengono caricati in un bucket Amazon S3. Durante tale periodo, l'agente AWS SCT continua a essere eseguito. L'agente quindi prende i dati su Amazon S3 e li copia nell'endpoint di destinazione.

Per ulteriori informazioni, consulta la pagina [Migrazione dei dati da un data warehouse locale ad Amazon Redshift](#) .

Utilizzo di unAWS SCT agente di estrazione dati conAWS DMS

Nello statoAWS SCT, è possibile trovare un agente di estrazione dati (`aws-schema-conversion-tool-extractor`) aiuta a semplificare la migrazione da Apache Cassandra ad Amazon DynamoDB. Cassandra e DynamoDB sono database NoSQL, ma differiscono per architettura di sistema e rappresentazione dei dati. È possibile utilizzare flussi di lavoro basati su procedure guidate inAWS SCTper automatizzare il processo di migrazione da Cassandra a DynamoDB.AWS SCTIntegrazione di conAWS Database Migration Service(AWS DMS) per eseguire la migrazione effettiva.

Per ulteriori informazioni, consulta la pagina [Migrazione dei dati da un data warehouse locale ad Amazon Redshift](#) .

Aumento dei livelli di registrazione durante l'utilizzoAWS SCTconAWS DMS

È possibile aumentare i livelli di registrazione quando si utilizzaAWS SCTconAWS DMS, ad esempio se hai bisogno di lavorare conAWS Support.

Dopo l'installazione AWS SCT e i driver richiesti, apri l'applicazione scegliendo il AWS SCT. Se viene visualizzata una notifica di aggiornamento, è possibile scegliere di aggiornare prima o dopo il completamento del progetto. Se si apre una finestra di progetto automatico, chiudere la finestra e creare manualmente un progetto.

Per aumentare i livelli di registrazione durante l'utilizzo AWS SCT con AWS DMS

1. Sull'impostazioni menu, scegli le impostazioni globali.
2. Nella finestra di impostazioni globali, scegli la registrazione di log.
3. Per la modalità di debug, scegli True.
4. Dal livello di messaggio sezione, è possibile modificare i seguenti tipi di log:
 - Generali
 - Loader
 - Parser
 - stampante
 - Resolver
 - Telemetria
 - Convertitore

Per impostazione predefinita, tutti i livelli di messaggio sono impostati sul Informazioni.

5. Scegli un livello di registrazione per qualsiasi tipo di livello di messaggio che desideri modificare:
 - Trace (registrazione più dettagliata)
 - Esegui il debug
 - Info
 - Attenzione
 - Errore (registrazione meno dettagliata)
 - Critico
 - Obbligatorio
6. Scegliere Applicare per modificare le impostazioni per il progetto.
7. Scegliere OK per chiudere la finestra di impostazioni globali.

Migrazione dei dati da un data warehouse locale ad Amazon Redshift

Puoi utilizzare un AWS SCT agente per estrarre dati dal tuo data warehouse locale e migrarli su Amazon Redshift. L'agente estrae i dati e li carica su Amazon S3 o, per le migrazioni su larga scala, su un dispositivo Edge. AWS Snowball Puoi quindi utilizzare un AWS SCT agente per copiare i dati su Amazon Redshift.

In alternativa, puoi utilizzare AWS Database Migration Service (AWS DMS) per migrare i dati su Amazon Redshift. Il vantaggio di AWS DMS è il supporto della replica continua (acquisizione dei dati di modifica). Tuttavia, per aumentare la velocità di migrazione dei dati, utilizza diversi AWS SCT agenti in parallelo. Secondo i nostri test, AWS SCT gli agenti migrano i dati più velocemente del AWS DMS 15-35 percento. La differenza di velocità è dovuta alla compressione dei dati, al supporto della migrazione delle partizioni di tabella in parallelo e alle diverse impostazioni di configurazione. Per ulteriori informazioni, consultare [Utilizzo di un database Amazon Redshift come destinazione per AWS Database Migration Service](#).

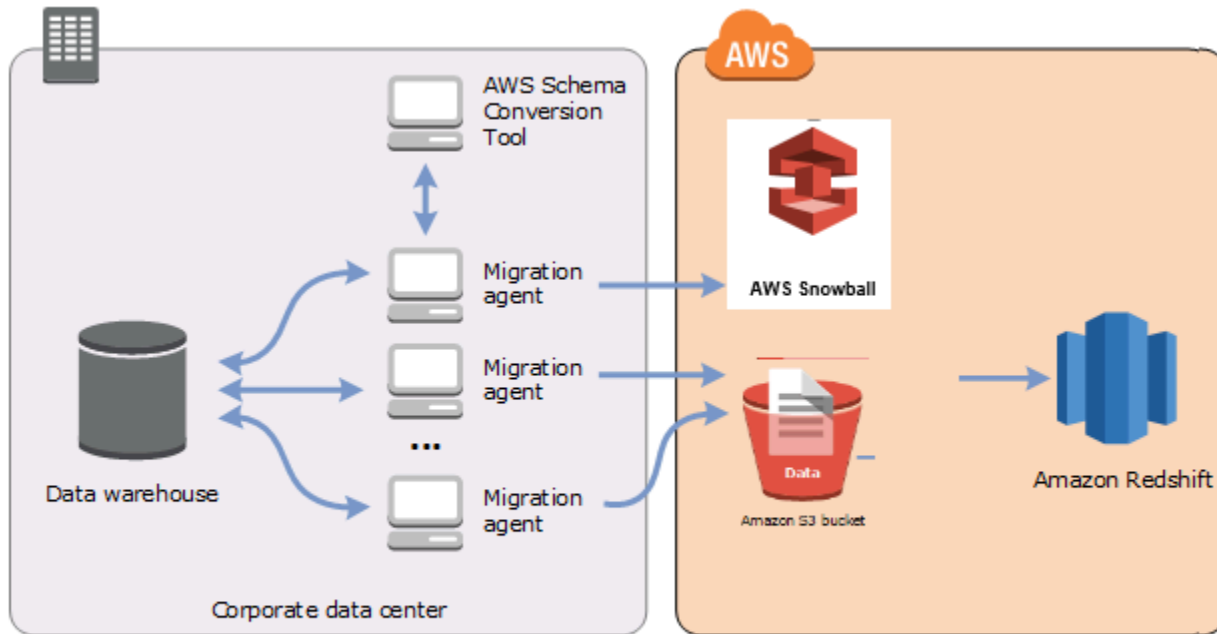
Amazon S3 è un servizio di storage e di recupero. Per archiviare un oggetto in Amazon S3, è necessario caricare il file da archiviare in un bucket Amazon S3. Durante il caricamento di un file, puoi impostare le autorizzazioni per l'oggetto e anche per eventuali metadata.

Migrazioni su larga scala

Le migrazioni di dati su larga scala possono includere molti terabyte di informazioni e possono essere rallentate dalle prestazioni della rete e dall'enorme quantità di dati da spostare. AWS Snowball Edge è un AWS servizio che puoi utilizzare per trasferire dati sul cloud a velocità elevate utilizzando un'appliance di proprietà. faster-than-network AWS Un dispositivo AWS Snowball Edge può contenere fino a 100 TB di dati. Utilizza una crittografia a 256 bit e un Trusted Platform Module (TPM) standard di settore per garantire la sicurezza e la completezza dei dati. chain-of-custody AWS SCT AWS Snowball funziona con dispositivi Edge.

Quando utilizzi AWS SCT un dispositivo AWS Snowball Edge, esegui la migrazione dei dati in due fasi. Innanzitutto, è necessario AWS SCT elaborare i dati localmente e quindi spostarli AWS Snowball sul dispositivo Edge. Quindi invii il dispositivo a AWS utilizzare il processo AWS Snowball Edge e quindi carichi AWS automaticamente i dati in un bucket Amazon S3. Successivamente, quando i dati sono disponibili su Amazon S3, li usi AWS SCT per migrare i dati su Amazon Redshift. Gli agenti di estrazione dei dati possono lavorare in background mentre AWS SCT sono chiusi.

Il seguente diagramma mostra lo scenario supportato.



Gli agenti di estrazione dei dati sono attualmente supportati per i seguenti data warehouse di origine:

- Azure Synapse Analytics
- BigQuery
- Database Greenplum (versione 4.3)
- Microsoft SQL Server (versione 2008 e successive)
- Netezza (versione 7.0.3 e successive)
- Oracle (versione 10 e successive)
- Snowflake (versione 3)
- Teradata (versione 13 e successive)
- Vertica (versione 7.2.2 e successive)

Puoi connetterti agli endpoint FIPS per Amazon Redshift se devi rispettare i requisiti di sicurezza del Federal Information Processing Standard (FIPS). Gli endpoint FIPS sono disponibili nelle seguenti regioni: AWS

- Regione Stati Uniti orientali (Virginia settentrionale) (redshift-fips.us-east-1.amazonaws.com)

- Regione Stati Uniti orientali (Ohio) (redshift-fips.us-east-2.amazonaws.com)
- Regione Stati Uniti occidentali (California settentrionale) (redshift-fips.us-west-1.amazonaws.com)
- Regione Stati Uniti occidentali (Oregon) (redshift-fips.us-west-2.amazonaws.com)

Utilizza le informazioni nei seguenti argomenti per scoprire come lavorare con gli agenti di estrazione dei dati.

Argomenti

- [Prerequisiti per l'utilizzo degli agenti di estrazione dei dati](#)
- [Installazione degli agenti di estrazione](#)
- [Configurazione degli agenti di estrazione](#)
- [Registrazione degli agenti di estrazione con AWS Schema Conversion Tool](#)
- [Nascondere e recuperare informazioni per un agente AWS SCT](#)
- [Creazione di regole di migrazione dei dati in AWS SCT](#)
- [Modifica dell'estrattore e copia le impostazioni dalle impostazioni del progetto](#)
- [Ordinamento dei dati prima della migrazione utilizzando AWS SCT](#)
- [Creazione, esecuzione e monitoraggio di un'attività di estrazione dei dati AWS SCT](#)
- [Esportazione e importazione di un'attività di estrazione dei dati AWS SCT](#)
- [Estrazione dei dati tramite un dispositivo AWS Snowball Edge](#)
- [Output dell'attività di estrazione dei dati](#)
- [Utilizzo del partizionamento virtuale con AWS Schema Conversion Tool](#)
- [Utilizzo del partizionamento nativo](#)
- [Migrazione dei LOB su Amazon Redshift](#)
- [Migliori pratiche e risoluzione dei problemi per gli agenti di estrazione dei dati](#)

Prerequisiti per l'utilizzo degli agenti di estrazione dei dati

Prima di lavorare con agenti di estrazione dati, aggiungi le autorizzazioni necessarie per Amazon Redshift come destinazione al tuo utente Amazon Redshift. Per ulteriori informazioni, consulta [Autorizzazioni per Amazon Redshift come obiettivo](#).

Quindi, archivia le informazioni sul bucket Amazon S3 e configura l'archivio di fiducia e chiavi Secure Sockets Layer (SSL).

Impostazioni Amazon S3

Dopo che i tuoi agenti hanno estratto i tuoi dati, li caricano nel tuo bucket Amazon S3. Prima di continuare, devi fornire le credenziali per connetterti al tuo AWS account e al tuo bucket Amazon S3. Memorizzi le credenziali e le informazioni sul bucket in un profilo nelle impostazioni globali dell'applicazione, quindi associ il profilo al tuo progetto. AWS SCT Se necessario, scegliete Impostazioni globali per creare un nuovo profilo. Per ulteriori informazioni, consulta [Archiviazione dei profili AWS di servizio in AWS SCT](#).

Per migrare i dati nel database Amazon Redshift di destinazione, AWS SCT l'agente di estrazione dei dati necessita dell'autorizzazione per accedere al bucket Amazon S3 per tuo conto. Per fornire questa autorizzazione, crea un utente AWS Identity and Access Management (IAM) con la seguente politica.

```
{
  "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"
    }
]
}
```

Nell'esempio precedente, sostituiscilo *bucket_name* con il nome del tuo bucket Amazon S3. Quindi, sostituiscilo *111122223333:user/DataExtractionAgentName* con il nome del tuo utente IAM.

Assumendo ruoli IAM

Per una maggiore sicurezza, puoi utilizzare i ruoli AWS Identity and Access Management (IAM) per accedere al tuo bucket Amazon S3. A tale scopo, crea un utente IAM per i tuoi agenti di estrazione dati senza alcuna autorizzazione. Quindi, crea un ruolo IAM che abiliti l'accesso ad Amazon S3 e specifica l'elenco di servizi e utenti che possono assumere questo ruolo. Per ulteriori informazioni, consulta [Ruoli IAM](#) nella Guida per l'utente IAM.

Per configurare i ruoli IAM per accedere al tuo bucket Amazon S3

1. Crea un nuovo utente IAM. Per le credenziali utente, scegli il tipo di accesso programmatico.
2. Configura l'ambiente host in modo che l'agente di estrazione dei dati possa assumere il ruolo che AWS SCT gli viene fornito. Assicurati che l'utente che hai configurato nel passaggio precedente consenta agli agenti di estrazione dei dati di utilizzare la catena di fornitori di credenziali. Per ulteriori informazioni, consulta [Utilizzo delle credenziali nella Guida](#) per gli AWS SDK for Java sviluppatori.
3. Crea un nuovo ruolo IAM con accesso al tuo bucket Amazon S3.
4. Modifica la sezione relativa alla fiducia di questo ruolo in modo che l'utente che hai creato in precedenza assuma il ruolo. Nell'esempio seguente, sostituiscilo *111122223333:user/DataExtractionAgentName* con il nome del tuo utente.

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/DataExtractionAgentName"
  },
  "Action": "sts:AssumeRole"
}
```

5. Modifica la sezione relativa alla fiducia di questo ruolo in modo che assuma il ruolo come `affidabileredshift.amazonaws.com`.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "redshift.amazonaws.com"
    ]
  },
  "Action": "sts:AssumeRole"
}
```

6. Collega questo ruolo al tuo cluster Amazon Redshift.

Ora puoi eseguire il tuo agente di estrazione dati in AWS SCT.

Quando si utilizza l'assunzione del ruolo IAM, la migrazione dei dati funziona nel modo seguente. L'agente di estrazione dei dati si avvia e ottiene le credenziali dell'utente utilizzando la catena di fornitori di credenziali. Successivamente, si crea un'attività di migrazione dei dati in AWS SCT, quindi si specifica il ruolo IAM che gli agenti di estrazione dei dati devono assumere e si avvia l'attività. AWS Security Token Service (AWS STS) genera credenziali temporanee per accedere ad Amazon S3. L'agente di estrazione dati utilizza queste credenziali per caricare dati su Amazon S3.

Quindi, AWS SCT fornisce ad Amazon Redshift il ruolo IAM. A sua volta, Amazon Redshift ottiene nuove credenziali temporanee per accedere AWS STS ad Amazon S3. Amazon Redshift utilizza queste credenziali per copiare i dati da Amazon S3 alla tabella Amazon Redshift.

Impostazioni di sicurezza

L'agente di estrazione AWS Schema Conversion Tool e gli agenti di estrazione possono comunicare tramite Secure Sockets Layer (SSL). Per abilitare SSL, configura un archivio della chiave e di trust.

Per configurare una comunicazione sicura con il tuo agente di estrazione

1. Avvia il. AWS Schema Conversion Tool
2. Apri il menu Impostazioni, quindi scegli Impostazioni globali. Si aprirà la finestra di dialogo Global Settings (Impostazioni globali).
3. Scegliere Sicurezza.
4. Scegli Genera fiducia e archivio chiavi oppure seleziona Seleziona archivio attendibile esistente.

Se scegli Generate trust e key store, specifichi il nome e la password per gli archivi trust e key e il percorso della posizione dei file generati. Utilizzerai questi file in un secondo momento.

Se scegli Select existing trust store, specifichi quindi la password e il nome del file per gli archivi trust e key. Utilizzerai questi file in un secondo momento.

5. Dopo aver specificato trust store e key store, scegliete OK per chiudere la finestra di dialogo delle impostazioni globali.

Configurazione dell'ambiente per gli agenti di estrazione dei dati

È possibile installare diversi agenti di estrazione dati su un singolo host. Tuttavia, si consiglia di eseguire un agente di estrazione dati su un host.

Per eseguire l'agente di estrazione dati, assicurati di utilizzare un host con almeno quattro vCPU e 32 GB di memoria. Inoltre, imposta la memoria minima disponibile su AWS SCT almeno quattro GB. Per ulteriori informazioni, consulta [Configurazione della memoria aggiuntiva](#).

La configurazione ottimale e il numero di host degli agenti dipendono dalla situazione specifica di ciascun cliente. Assicurati di considerare fattori quali la quantità di dati da migrare, la larghezza di banda della rete, il tempo necessario per estrarre i dati e così via. È possibile eseguire prima un proof of concept (PoC), quindi configurare gli agenti e gli host di estrazione dei dati in base ai risultati di questo PoC.

Installazione degli agenti di estrazione

È consigliabile installare più agenti di estrazione sui singoli computer, separati dal computer su cui è in esecuzione AWS Schema Conversion Tool.

Gli agenti di estrazione sono attualmente supportati per i seguenti sistemi operativi:

- Microsoft Windows
- Red Hat Enterprise Linux (RHEL) 6.0
- Ubuntu Linux (versione 14.04 e successive)

Utilizza la procedura seguente per installare gli agenti di estrazione. Ripeti questa procedura per ogni computer su cui si desidera installare un agente di estrazione.

Per installare un agente di estrazione

1. Se non avete ancora scaricato il file di AWS SCT installazione, seguite le istruzioni riportate [Installazione, verifica e aggiornamento AWS SCT](#) per scaricarlo. Il file.zip che contiene il file di AWS SCT installazione contiene anche il file di installazione dell'agente di estrazione.
2. Scarica e installa l'ultima versione di Amazon Corretto 11. Per ulteriori informazioni, consulta [Download per Amazon Corretto 11](#) nella Guida per l'utente di Amazon Corretto 11.
3. Individua il file del programma di installazione del tuo agente di estrazione in una sottocartella denominata agenti. Per ogni sistema operativo, il file corretto per installare l'agente di estrazione è riportato di seguito.

Sistema operativo	Nome 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. Installa l'agente di estrazione su un computer separato copiando il file di installazione sul nuovo computer.
5. Esegui il file del programma di installazione. Utilizza le istruzioni per il tuo sistema operativo riportate di seguito.

Sistema operativo	Istruzioni di installazione
Microsoft Windows	Fai doppio clic sul file per eseguire il programma di installazione.
RHEL	<p>Esegui i seguenti comandi nella cartella in cui hai scaricato o spostato il 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>Esegui i seguenti comandi nella cartella in cui hai scaricato o spostato il 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>

- Scegliete Avanti, accettate il contratto di licenza e scegliete Avanti.
- Inserisci il percorso per installare l'agente di estrazione AWS SCT dei dati e scegli Avanti.
- Scegli Installa per installare l'agente di estrazione dati.

AWS SCT installa il tuo agente di estrazione dati. Per completare l'installazione, configura l'agente di estrazione dati. AWS SCT avvia automaticamente il programma di configurazione. Per ulteriori informazioni, consulta [Configurazione degli agenti di estrazione](#).

- Scegli Fine per chiudere la procedura guidata di installazione dopo aver configurato l'agente di estrazione dei dati.

Configurazione degli agenti di estrazione

Utilizza la procedura seguente per configurare gli agenti di estrazione. Ripeti questa procedura su ogni computer su cui è installato un agente di estrazione.

Per configurare l'agente di estrazione

- Avvia il programma di configurazione:

- In Windows, AWS SCT avvia automaticamente il programma di configurazione durante l'installazione di un agente di estrazione dati.

Se necessario, è possibile avviare il programma di installazione manualmente. A tale scopo, esegui il `ConfigAgent.bat` file in Windows. Puoi trovare questo file nella cartella in cui hai installato l'agente.

- In RHEL e Ubuntu, esegui il `sct-extractor-setup.sh` file dalla posizione in cui hai installato l'agente.

Il programma di installazione richiede informazioni. Per ogni richiesta, appare un valore predefinito.

2. Accettate il valore predefinito a ogni richiesta o immettete un nuovo valore.

Specificare le seguenti informazioni:

- Per Porta di ascolto, inserite il numero di porta su cui l'agente è in ascolto.
- Per Aggiungi un fornitore di origine, inserisci sì, quindi inserisci la piattaforma di data warehouse di origine.
- Per il driver JDBC, inserisci la posizione in cui hai installato i driver JDBC.
- Per Working folder, inserisci il percorso in cui l'agente di estrazione dei AWS SCT dati memorizzerà i dati estratti. La cartella di lavoro può essere su un computer diverso dall'agente e una singola cartella di lavoro può essere condivisa da più agenti su diversi computer.
- Per Abilita la comunicazione SSL, inserisci yes.
- Per Archivio chiavi, inserisci la posizione del file dell'archivio delle chiavi.
- Per Password dell'archivio chiavi, inserisci la password per l'archivio chiavi.
- Per Abilita l'autenticazione SSL del client, inserisci yes.
- Per Trust store, inserisci la posizione del file Trust Store.
- Per la password del Trust Store, inserisci la password per il Trust Store.

Il programma di installazione aggiorna il file delle impostazioni per l'agente di estrazione. Il file delle impostazioni è denominato `settings.properties` e si trova nella stessa posizione in cui hai installato l'agente di estrazione.

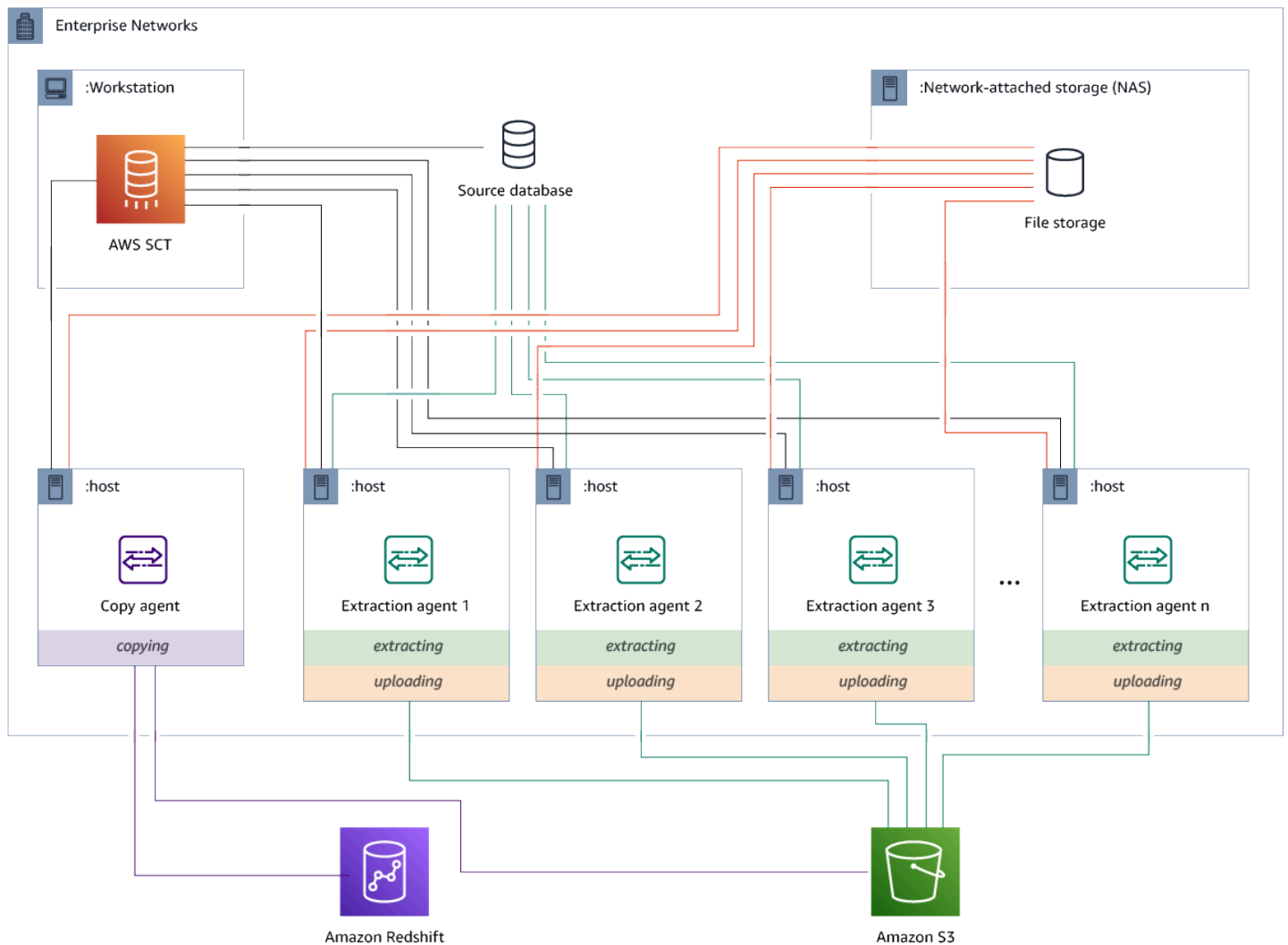
Di seguito è riportato un esempio di file delle impostazioni.

```
$ 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
```

Per modificare le impostazioni di configurazione, è possibile modificare il `settings.properties` file utilizzando un editor di testo o eseguire nuovamente la configurazione dell'agente.

Installazione e configurazione degli agenti di estrazione con agenti di copia dedicati

È possibile installare agenti di estrazione in una configurazione con storage condiviso e un agente di copia dedicato. Il diagramma seguente illustra questo scenario.



Questa configurazione può essere utile quando un server di database di origine supporta fino a 120 connessioni e la rete dispone di un ampio spazio di archiviazione collegato. Utilizzate la procedura seguente per configurare gli agenti di estrazione con un agente di copia dedicato.

Per installare e configurare agenti di estrazione e un agente di copia dedicato

1. Assicuratevi che la directory di lavoro di tutti gli agenti di estrazione utilizzi la stessa cartella sullo storage condiviso.
2. Installa gli agenti di estrazione seguendo la procedura riportata di seguito. [Installazione degli agenti di estrazione](#)
3. Configura gli agenti di estrazione seguendo i passaggi indicati [Configurazione degli agenti di estrazione](#), ma specifica solo il driver JDBC di origine.

4. Configura un agente di copia dedicato seguendo i passaggi indicati [Configurazione degli agenti di estrazione](#), ma specifica solo un driver JDBC Amazon Redshift.

Avvio degli agenti di estrazione

Utilizza la procedura seguente per avviare gli agenti di estrazione. Ripeti questa procedura su ogni computer su cui è installato un agente di estrazione.

Gli agenti di estrazione fungono da ascoltatori. Quando avvii un agente con questa procedura, l'agente avvia l'ascolto delle relative istruzioni. Puoi inviare agli agenti le istruzioni per estrarre i dati provenienti dal data warehouse in una sezione successiva.

Per avviare l'agente di estrazione

- Nel computer in cui è installato l'agente di estrazione, esegui il comando elencato di seguito per il tuo sistema operativo.

Sistema operativo	Comando di avvio
Microsoft Windows	Fai doppio clic sul file batch <code>StartAgent.bat</code> .
RHEL	Esegui il comando seguente nel percorso per la cartella in cui hai installato l'agente: <code>sudo initctl <i>start</i> sct-extractor</code>
Ubuntu Linux	Esegui il comando seguente nel percorso per la cartella in cui hai installato l'agente. Utilizza il comando appropriato alla tua versione di Ubuntu. Ubuntu 14.04: <code>sudo initctl <i>start</i> sct-extractor</code> Ubuntu 15.04 e versioni successive: <code>sudo systemctl <i>start</i> sct-extractor</code>

Per controllare lo stato dell'agente, esegui lo stesso comando, ma sostituisci `start` con `status`.

Per arrestare un agente, esegui lo stesso comando, ma sostituisci `start` con `stop`.

Registrazione degli agenti di estrazione con AWS Schema Conversion Tool

Gestisci i tuoi agenti di estrazione utilizzando AWS SCT. Gli agenti di estrazione fungono da ascoltatori. Quando ricevono istruzioni da AWS SCT, estraggono i dati dal tuo data warehouse.

Utilizzate la seguente procedura per registrare gli agenti di estrazione nel AWS SCT progetto.

Per registrare un agente di estrazione

1. AWS Schema Conversion Tool Avviate e aprite un progetto.
2. Apri il menu Visualizza, quindi scegli Data Migration view (altro). Viene visualizzata la scheda Agents (Agenti). Se hai già degli agenti registrati, li AWS SCT visualizza in una griglia nella parte superiore della scheda.
3. Scegli Registrati.

Dopo aver registrato un agente per un AWS SCT progetto, non puoi registrare lo stesso agente per un progetto diverso. Se non utilizzi più un agente in un AWS SCT progetto, puoi annullarne la registrazione. Puoi registrarlo con un progetto diverso.

4. Scegli Redshift Data Agent, quindi scegli OK.
5. Inserisci le tue informazioni nella scheda Connessione della finestra di dialogo:
 - a. In Descrizione, inserisci una descrizione dell'agente.
 - b. Per Nome host, inserire il nome host o l'indirizzo IP del computer dell'agente.
 - c. In Porta, inserisci il numero di porta su cui l'agente è in ascolto.
 - d. Scegli Registra per registrare l'agente nel tuo AWS SCT progetto.
6. Ripeti i passaggi precedenti per registrare più agenti con il tuo progetto AWS SCT .

Nascondere e recuperare informazioni per un agente AWS SCT

Un AWS SCT agente crittografa una quantità significativa di informazioni, ad esempio le password degli archivi di fiducia delle chiavi degli utenti, gli account del database, le informazioni sugli AWS account e elementi simili. Lo fa utilizzando un file speciale chiamato `seed.dat`. Per impostazione predefinita, l'agente crea questo file nella cartella di lavoro dell'utente che configura per primo l'agente.

Poiché utenti diversi possono configurare ed eseguire l'agente, il percorso per `seed.dat` viene memorizzato nel parametro `{extractor.private.folder}` del file `settings.properties`. Quando l'agente viene avviato, può usare questo percorso per trovare il file `seed.dat` per accedere alle informazioni relative all'archivio di `trust-della` chiave per il database su cui agisce.

Potresti dover recuperare le password che un agente ha archiviato in questi casi:

- Se l'utente perde il `seed.dat` file e la posizione e la porta dell' AWS SCT agente non sono cambiate.
- Se l'utente perde il `seed.dat` file e la posizione e la porta dell' AWS SCT agente sono cambiate. In questo caso, la modifica in genere si verifica perché l'agente è stato migrato a un altro host o la porta e le informazioni contenute nel `seed.dat` file non sono più valide.

In questi casi, se un agente viene avviato senza SSL, si avvia e quindi accede allo storage dell'agente creato in precedenza. Quindi va nello stato `Waiting for recovery` (In attesa di ripristino)..

Tuttavia, in questi casi, se un agente viene avviato con SSL non puoi riavviarlo. Questo perché l'agente non è in grado di decriptare le password archiviate nel file `settings.properties`. In questo tipo di avvio, l'agente non riesce ad avviarsi. Un errore simile al seguente è scritto nel log: "L'agente non è stato in grado di avviarsi con la modalità SSL abilitata. Si prega di riconfigurare l'agente. Motivo: la password per il keystore non è corretta."

Per risolvere questo problema, crea un nuovo agente e configuralo in modo da utilizzare le password esistenti per accedere ai certificati SSL. A tale scopo, procedi come indicato di seguito.

Dopo aver eseguito questa procedura, l'agente dovrebbe avviarsi e passare allo stato `In attesa di ripristino`. AWS SCT invia automaticamente le password necessarie a un agente nello stato `Waiting for recovery`. Quando l'agente ha le password, riavvia qualsiasi attività. Non sono richieste ulteriori azioni da parte dell'utente da AWS SCT parte dell'utente.

Per riconfigurare l'agente e ripristinare le password per l'accesso ai certificati SSL

1. Installa un nuovo AWS SCT agente ed esegui la configurazione.
2. Cambia la proprietà `agent.name` nel file `instance.properties` per il nome dell'agente per cui lo storage è stato creato, per far sì che il nuovo agente sia compatibile con lo storage dell'agente esistente.

Il file `instance.properties` viene archiviato nella cartella privata dell'agente, denominata utilizzando la seguente convenzione: `{output.folder}\dmt\{hostname}_{portNumber}\`.

3. Cambia il nome di `{output.folder}` in quello della cartella di output dell'agente precedente.

A questo punto, AWS SCT sta ancora cercando di accedere al vecchio estrattore sul vecchio host e sulla vecchia porta. Di conseguenza, l'estrattore inaccessibile ottiene lo stato FAILED. Puoi modificare l'host e la porta.

4. Modifica l'host, la porta (o entrambi) dell'agente precedente utilizzando il comando Modifica per reindirizzare il flusso di richiesta al nuovo agente.

Quando AWS SCT può eseguire il ping del nuovo agente, AWS SCT riceve lo stato In attesa di ripristino dall'agente. AWS SCT quindi recupera automaticamente le password dell'agente.

Ogni agente che funziona con lo storage dell'agente aggiorna un file speciale chiamato `storage.lock` posizionato in `{output.folder}\{agentName}\storage\`. Questo file contiene l'ID di rete dell'agente e il tempo durante cui lo storage è bloccato. Quando l'agente funziona con lo storage dell'agente, aggiorna il file `storage.lock` ed estende la decorrenza dello storage di 10 minuti ogni 5 minuti. Nessuna altra istanza può funzionare con lo storage di questo agente prima della scadenza della decorrenza.

Creazione di regole di migrazione dei dati in AWS SCT

Prima di estrarre i dati con AWS Schema Conversion Tool, puoi impostare filtri che riducano la quantità di dati che estrai. È possibile creare regole di migrazione dei dati utilizzando WHERE clausole per ridurre i dati estratti. Ad esempio, puoi scrivere una clausola WHERE che seleziona i dati da un'unica tabella.

Puoi creare regole di migrazione dei dati e salvare i filtri come parte del tuo progetto. Con il progetto aperto, utilizzate la procedura seguente per creare regole di migrazione dei dati.

Per creare regole di migrazione dei dati

1. Apri il menu Visualizza, quindi scegli Data Migration view (altro).
2. Scegli Regole di migrazione dei dati, quindi scegli Aggiungi nuova regola.
3. Configura la tua regola di migrazione dei dati:

- a. In Nome, inserisci un nome per la regola di migrazione dei dati.
 - b. Per Where schema name is like, inserisci un filtro da applicare agli schemi. In questo filtro, una clausola WHERE viene valutata utilizzando una clausola LIKE. Per scegliere uno schema, inserisci un nome esatto dello schema. Per scegliere più schemi, usa il carattere «%» come jolly per abbinare un numero qualsiasi di caratteri nel nome dello schema.
 - c. Ad esempio, inserisci un filtro da applicare alle tabelle. In questo filtro, una clausola WHERE viene valutata utilizzando una clausola LIKE. Per scegliere una tabella, inserisci un nome esatto. Per scegliere più tabelle, usa il carattere «%» come jolly per abbinare un numero qualsiasi di caratteri nel nome della tabella.
 - d. Per la clausola Where, inserisci una **WHERE** clausola per filtrare i dati.
4. Dopo aver configurato il filtro, selezionare Salva per salvare il filtro o Annulla per annullare le modifiche.
 5. Dopo aver aggiunto, modificato ed eliminato i filtri, scegli Salva tutto per salvare tutte le modifiche.

Per disattivare un filtro senza eliminarlo, utilizza l'icona di attivazione/disattivazione. Per duplicare un filtro esistente, utilizza l'icona Copia. Per eliminare un filtro esistente, utilizza l'icona Elimina. Per salvare le modifiche apportate ai filtri, scegli Salva tutto.

Modifica dell'estrattore e copia le impostazioni dalle impostazioni del progetto

Dalla finestra delle impostazioni del progetto in AWS SCT, puoi scegliere le impostazioni per gli agenti di estrazione dei dati e il comando Amazon RedshiftCOPY.

Per scegliere queste impostazioni, scegli Impostazioni, Impostazioni del progetto, quindi scegli Migrazione dei dati. Qui puoi modificare le impostazioni di estrazione, le impostazioni di Amazon S3 e le impostazioni di copia.

Utilizza le istruzioni nella tabella seguente per fornire le informazioni per le impostazioni di estrazione.

Per questo parametro	Esegui questa operazione
Formato di compressione	Specificate il formato di compressione dei file di input. Scegliete una delle seguenti opzioni: GZIP, BZIP2, ZSTD o Nessuna compressione.
Carattere delimitatore	Specificate il carattere ASCII che separa i campi nei file di input. I caratteri non stampabili non sono supportati.
Valore NULL come stringa	Attiva questa opzione se i dati includono un terminatore nullo. Se questa opzione è disattivata, il COPY comando Amazon Redshift considera null come la fine del record e termina il processo di caricamento.
Strategia di ordinamento	Usa l'ordinamento per riavviare l'estrazione dal punto di errore. Scegliete una delle seguenti strategie di ordinamento: Usa l'ordinamento dopo il primo errore (consigliato), Usa l'ordinamento se possibile o Non usare mai l'ordinamento. Per ulteriori informazioni, consulta the section called "Ordinamento dei dati" .
Schema temporaneo di origine	Immettete il nome dello schema nel database di origine, in cui l'agente di estrazione può creare gli oggetti temporanei.
Dimensione del file in uscita (in MB)	Inserisci la dimensione, in MB, dei file caricati su Amazon S3.
Dimensione del file Snowball out (in MB)	Inserisci la dimensione, in MB, dei file caricati su AWS Snowball. Le dimensioni dei file possono essere comprese tra 1 e 1.000 MB.
Utilizza il partizionamento automatico. Per Greenplum e Netezza, inserisci la dimensione minima delle tabelle supportate (in megabyte)	Attiva questa opzione per utilizzare il partizionamento delle tabelle, quindi inserisci la dimensione delle tabelle da partizionare per i database di origine Greenplum e Netezza. Per le migrazioni da Oracle ad Amazon Redshift, puoi lasciare vuoto questo campo perché AWS SCT crea sottoattività per tutte le tabelle partizionate.
Estrai LOB	Attiva questa opzione per estrarre oggetti di grandi dimensioni (LOB) dal database di origine. I LOB includono BLOB, CLOB,

Per questo parametro	Esegui questa operazione
	NCLOB, file XML e così via. Per ogni LOB, gli agenti di estrazione e creano un file di dati. AWS SCT
Cartella LOB con bucket Amazon S3	Inserisci la posizione degli agenti di AWS SCT estrazione in cui archiviare i LOB.
Applica RTRIM alle colonne di stringhe	Attiva questa opzione per tagliare un set di caratteri specificato dalla fine delle stringhe estratte.
Conserva i file localment e dopo il caricamento su Amazon S3	Attiva questa opzione per conservare i file sul computer locale dopo che gli agenti di estrazione dei dati li hanno caricati su Amazon S3.

Utilizza le istruzioni nella tabella seguente per fornire le informazioni per le impostazioni di Amazon S3.

Per questo parametro	Esegui questa operazione
Usa un proxy	Attiva questa opzione per utilizzare un server proxy per caricare dati su Amazon S3. Quindi scegli il protocollo di trasferimento dati, inserisci il nome host, la porta, il nome utente e la password.
Endpoint type (Tipo di endpoint)	Scegli FIPS per utilizzare l'endpoint FIPS (Federal Information Processing Standard). Scegli VPCE per utilizzare l'endpoint del cloud privato virtuale (VPC). Quindi, per l'endpoint VPC, inserisci il Domain Name System (DNS) del tuo endpoint VPC.
Conserva i file su Amazon S3 dopo averli copiati su Amazon Redshift	Attiva questa opzione per conservare i file estratti su Amazon S3 dopo averli copiati su Amazon Redshift.

Utilizza le istruzioni nella tabella seguente per fornire le informazioni per le impostazioni di copia.

Per questo parametro	Esegui questa operazione
Numero massimo di errori	Immettere il numero di errori di caricamento. Dopo che l'operazione raggiunge questo limite, gli agenti di estrazione AWS SCT dei dati terminano il processo di caricamento dei dati. Il valore predefinito è 0, il che significa che gli agenti di estrazione AWS SCT dei dati continuano il caricamento dei dati indipendentemente dagli errori.
Sostituisce i caratteri UTF-8 non validi	Attiva questa opzione per sostituire i caratteri UTF-8 non validi con il carattere specificato e continuare l'operazione di caricamento dei dati.
Usa vuoto come valore nullo	Attiva questa opzione per caricare campi vuoti costituiti da caratteri di spazio bianco come nulli.
Usa vuoto come valore nullo	Attiva questa opzione per caricare campi vuoti CHAR e VARCHAR campi come nulli.
Tronca le colonne	Attiva questa opzione per troncatura i dati in colonne per adattarli alle specifiche del tipo di dati.
Compressione automatica	Attiva questa opzione per applicare la codifica di compressione durante un'operazione di copia.
Aggiornamento automatico delle statistiche	Attiva questa opzione per aggiornare le statistiche al termine di un'operazione di copia.
Controlla il file prima del caricamento	Attiva questa opzione per convalidare i file di dati prima di caricarli su Amazon Redshift.

Ordinamento dei dati prima della migrazione utilizzando AWS SCT

L'ordinamento dei dati prima della migrazione AWS SCT offre alcuni vantaggi. Se si ordinano prima i dati, AWS SCT è possibile riavviare l'agente di estrazione dall'ultimo punto salvato dopo un errore. Inoltre, se stai migrando i dati su Amazon Redshift e li ordini prima AWS SCT, puoi inserirli in Amazon Redshift più velocemente.

Questi vantaggi riguardano il modo in cui vengono AWS SCT create le query di estrazione dei dati. In alcuni casi, AWS SCT utilizza la funzione analitica DENSE_RANK in queste query. Tuttavia, DENSE_RANK può utilizzare molto tempo e risorse del server per ordinare il set di dati risultante dall'estrazione, quindi se può funzionare senza di esso, funziona. AWS SCT

Per ordinare i dati prima della migrazione, utilizzare AWS SCT

1. Aprire un AWS SCT progetto.
2. Apri il menu contestuale (fai clic con il pulsante destro del mouse) per l'oggetto, quindi scegli Crea attività locale.
3. Scegli la scheda Avanzate e, per Strategia di ordinamento, scegli un'opzione:
 - Non utilizzare mai l'ordinamento: l'agente di estrazione non utilizza la funzione analitica DENSE_RANK e si riavvia dall'inizio in caso di errore.
 - Se possibile, utilizzate l'ordinamento: l'agente di estrazione utilizza DENSE_RANK se la tabella ha una chiave primaria o un vincolo univoco.
 - Utilizza l'ordinamento dopo il primo errore (consigliato): l'agente di estrazione tenta innanzitutto di ottenere i dati senza utilizzare DENSE_RANK. Se il primo tentativo ha esito negativo, l'agente di estrazione ricostruisce la query utilizzando DENSE_RANK e conserva la propria posizione in caso di guasto.

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

- Impostare parametri aggiuntivi come descritto di seguito, quindi scegli Crea per creare la tua attività di estrazione dei dati.

Creazione, esecuzione e monitoraggio di un'attività di estrazione dei dati AWS SCT

Utilizza le seguenti procedure per creare, eseguire e monitorare le attività di estrazione dei dati.

Per assegnare le attività agli agenti e migrare i dati

1. In AWS Schema Conversion Tool, dopo aver convertito lo schema, scegli una o più tabelle dal pannello di sinistra del progetto.

Puoi scegliere tutte le tabelle, ma è un'azione che sconsigliamo per motivi di prestazioni. È consigliabile creare più attività per più tabelle in base alle dimensioni delle tabelle del data warehouse.

2. Apri il menu contestuale (fai clic con il pulsante destro del mouse) per ogni tabella, quindi scegli Crea attività. Viene visualizzata la finestra di dialogo Crea attività locale.
3. In Nome attività, inserisci un nome per l'attività.
4. Per la modalità di migrazione, scegli una delle seguenti opzioni:
 - Solo estrazione: estrai i dati e salvali nelle cartelle di lavoro locali.
 - Estrazione e caricamento: estrai i dati e caricali su Amazon S3.
 - Estrai, carica e copia: estrai i dati, carica i dati su Amazon S3 e copiali nel tuo data warehouse Amazon Redshift.
5. Per il tipo di crittografia, scegli una delle seguenti opzioni:
 - NESSUNA: disattiva la crittografia dei dati per l'intero processo di migrazione dei dati.
 - CSE_SK: utilizza la crittografia lato client con una chiave simmetrica per migrare i dati. AWS SCT genera automaticamente chiavi di crittografia e le trasmette agli agenti di estrazione dei dati utilizzando Secure Sockets Layer (SSL). AWS SCT non crittografa oggetti di grandi dimensioni (LOB) durante la migrazione dei dati.
6. Scegliere Extract LOBs (Estrai LOB) per estrarre gli oggetti di grandi dimensioni. Se non è necessario estrarre gli oggetti di grandi dimensioni, è possibile deselezionare la casella di controllo. In questo modo è possibile ridurre la quantità di dati estratti.
7. Per visualizzare informazioni dettagliate su un'attività, scegli Abilita registrazione delle attività. Puoi utilizzare i log dell'attività per il debug dei problemi.

Se abiliti il log delle attività, scegli il livello di dettaglio che desideri visualizzare. I livelli sono i seguenti, con ogni livello che include tutti i messaggi del livello precedente:

- ERROR— La minima quantità di dettagli.
- WARNING
- INFO

- DEBUG
 - TRACE— La massima quantità di dettagli.
8. Per esportare i dati da BigQuery, AWS SCT utilizza la cartella bucket di Google Cloud Storage. In questa cartella, gli agenti di estrazione dei dati archiviano i dati di origine.

Per inserire il percorso della cartella bucket di Google Cloud Storage, scegli Avanzate. Per la cartella bucket di Google CS, inserisci il nome del bucket e il nome della cartella.

9. Per assumere un ruolo come utente del tuo agente di estrazione dati, scegli le impostazioni di Amazon S3. Per il ruolo IAM, inserisci il nome del ruolo da utilizzare. Per Regione, scegli il ruolo Regione AWS per questo ruolo.
10. Scegli Test task per verificare che sia possibile connettersi alla cartella di lavoro, al bucket Amazon S3 e al data warehouse Amazon Redshift. La verifica dipende dalla modalità di migrazione scelta.
11. Scegli Create (crea) per creare l'attività.
12. Ripeti i passaggi precedenti per creare le attività per tutti i dati che desideri migrare.

Per eseguire e monitorare le attività

1. Per View, scegli Data Migration view. Viene visualizzata la scheda Agents (Agenti).
2. Selezionare la scheda Tasks (Attività). Le tue attività compaiono nella griglia nella parte superiore come illustrato di seguito. Puoi visualizzare lo stato di un'attività nella griglia superiore e lo stato delle sottoattività nella griglia inferiore.

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%

Download log All migration Tasks Resume Stop Restart Reset Delete Replace Refresh all Refresh selected

Properties Processing details

Extracting
Data
100%

3. Scegli un'attività nella griglia superiore ed espandila. A seconda della modalità di migrazione scelta, vedrai l'attività divisa in Extract (Estrai), Carica e Copy (Copia).
4. Scegliere Start (Avvia) per un'attività per avviarla. Puoi monitorare lo stato delle attività durante il loro lavoro. Le sottoattività vengono eseguite in parallelo. Anche l'estrazione, il caricamento e la copia vengono eseguiti in parallelo.
5. Se hai abilitato la creazione di log quando configuri l'attività, puoi visualizzare i log:
 - a. Scegli Scarica registro. Viene visualizzato un messaggio con il nome della cartella che contiene i file di log. Elimina il messaggio.
 - b. Un link viene visualizzato nella scheda Task details (Dettagli dell'attività). Scegli il link per aprire la cartella che contiene i file di log.

Puoi chiudere AWS SCT e gli agenti e le attività continueranno a funzionare. Puoi riaprirle in un AWS SCT secondo momento per controllare lo stato delle attività e visualizzare i registri delle attività.

È possibile salvare le attività di estrazione dei dati sul disco locale e ripristinarle sullo stesso o su un altro progetto utilizzando l'esportazione e l'importazione. Per esportare un'attività, assicurati di avere almeno un'attività di estrazione creata in un progetto. È possibile importare una singola attività di estrazione o tutte le attività create nel progetto.

Quando esporti un'attività di estrazione, AWS SCT crea un .xml file separato per quell'attività. Il .xml file memorizza le informazioni sui metadati dell'attività, come le proprietà, la descrizione e le sottoattività. Il .xml file non contiene informazioni sull'elaborazione di un'operazione di estrazione. Informazioni come le seguenti vengono ricreate quando l'attività viene importata:

- Avanzamento dell'attività
- Sottoattività e stati dello stadio
- Distribuzione degli agenti di estrazione per sottoattività e fasi
- ID delle attività e delle sottoattività
- Nome attività

Esportazione e importazione di un'attività di estrazione dei dati AWS SCT

È possibile salvare rapidamente un'attività esistente da un progetto e ripristinarla in un altro progetto (o nello stesso progetto) utilizzando l' AWS SCT esportazione e l'importazione. Utilizzate la seguente procedura per esportare e importare le attività di estrazione dei dati.

Per esportare e importare un'attività di estrazione dei dati

1. Per Visualizza, scegli la visualizzazione Data Migration. Viene visualizzata la scheda Agents (Agenti).
2. Selezionare la scheda Tasks (Attività). Le tue attività sono elencate nella griglia visualizzata.
3. Scegli i tre punti allineati verticalmente (icona con i puntini di sospensione) situati nell'angolo in basso a destra sotto l'elenco delle attività.
4. Scegli Esporta attività dal menu pop-up.
5. Scegli la cartella in cui desideri AWS SCT inserire il .xml file di esportazione dell'attività.

AWS SCT crea il file di esportazione delle attività con il formato del nome file di **TASK-DESCRIPTION_TASK-ID.xml**.

6. Scegli i tre punti allineati verticalmente (icona con i puntini di sospensione) in basso a destra sotto l'elenco delle attività.
7. Scegli Importa attività dal menu pop-up.

È possibile importare un'attività di estrazione in un progetto collegato al database di origine e il progetto ha almeno un agente di estrazione registrato attivo.

8. Seleziona il .xml file per l'attività di estrazione che hai esportato.

AWS SCT ottiene i parametri dell'attività di estrazione dal file, crea l'attività e la aggiunge agli agenti di estrazione.

9. Ripeti questi passaggi per esportare e importare altre attività di estrazione dei dati.

Al termine di questo processo, l'esportazione e l'importazione sono complete e le attività di estrazione dei dati sono pronte per l'uso.

Estrazione dei dati tramite un dispositivo AWS Snowball Edge

Il processo di utilizzo AWS SCT di AWS Snowball Edge prevede diversi passaggi. La migrazione prevede un'attività locale, in cui AWS SCT utilizza un agente di estrazione dati per spostare i dati sul dispositivo AWS Snowball Edge, quindi un'azione intermedia in cui AWS copia i dati dal dispositivo AWS Snowball Edge a un bucket Amazon S3. Il processo termina il AWS SCT caricamento dei dati dal bucket Amazon S3 su Amazon Redshift.

Le sezioni che seguono questa panoramica forniscono una step-by-step guida per ciascuna di queste attività. La procedura presuppone che sia stato AWS SCT installato e configurato e registrato un agente di estrazione dati su un computer dedicato.

Esegui i passaggi seguenti per migrare i dati da un data store locale a un AWS data store utilizzando AWS Snowball Edge.

1. Crea un job AWS Snowball Edge utilizzando la AWS Snowball console.
2. Sblocca il dispositivo AWS Snowball Edge utilizzando il computer Linux locale dedicato.
3. Crea un nuovo progetto in AWS SCT.
4. Installa e configura i tuoi agenti di estrazione dei dati.
5. Crea e imposta le autorizzazioni per il bucket Amazon S3 da utilizzare.
6. Importa un AWS Snowball lavoro nel tuo AWS SCT progetto.
7. Registra il tuo agente di estrazione dati in AWS SCT.
8. Crea un'attività locale in AWS SCT.
9. Esegui e monitora l'attività di migrazione dei dati in AWS SCT.

tep-by-step Procedure S per la migrazione dei dati tramite AWS SCT Edge AWS Snowball

Le sezioni seguenti forniscono informazioni dettagliate sulle fasi della migrazione.

Passaggio 1: creare un lavoro AWS Snowball Edge

Crea un AWS Snowball lavoro seguendo i passaggi descritti nella sezione [Creating an AWS Snowball Edge Job nella AWS Snowball Edge Developer Guide](#).

Passaggio 2: sblocca il dispositivo AWS Snowball Edge

Esegui i comandi che sbloccano e forniscono le credenziali al dispositivo Snowball Edge dal computer su cui hai installato AWS DMS l'agente. Eseguendo questi comandi, puoi essere sicuro che la chiamata dell' AWS DMS agente si connetta al dispositivo AWS Snowball Edge. Per ulteriori informazioni sullo sblocco del dispositivo AWS Snowball Edge, consulta [Sblocco di Snowball Edge](#).

```
aws s3 ls s3://<bucket-name> --profile <Snowball Edge profile> --endpoint http://<Snowball IP>:8080 --recursive
```

Fase 3: Creare un nuovo progetto AWS SCT

Quindi, crea un nuovo AWS SCT progetto.

Per creare un nuovo progetto in AWS SCT

1. Avvia il AWS Schema Conversion Tool. Nel menu File, scegli Nuovo progetto. Viene visualizzata la finestra di dialogo Nuovo progetto.
2. Immetti un nome per il progetto, che è memorizzato localmente nel computer.
3. Immetti l'ubicazione del file del progetto locale.
4. Scegliete OK per creare il AWS SCT progetto.
5. Scegli Aggiungi sorgente per aggiungere un nuovo database di origine al tuo AWS SCT progetto.
6. Scegli Aggiungi destinazione per aggiungere una nuova piattaforma di destinazione al tuo AWS SCT progetto.
7. Scegli lo schema del database di origine nel pannello di sinistra.
8. Nel pannello di destra, specifica la piattaforma di database di destinazione per lo schema di origine selezionato.

9. Scegli Crea mappatura. Questo pulsante diventa attivo dopo aver scelto lo schema del database di origine e la piattaforma di database di destinazione.

Passaggio 4: installa e configura l'agente di estrazione dei dati

AWS SCT utilizza un agente di estrazione dati per migrare i dati su Amazon Redshift. Il file.zip che hai scaricato per l'installazione include il file di installazione AWS SCT dell'agente di estrazione. È possibile installare l'agente di estrazione dei dati in Windows, Red Hat Enterprise Linux o Ubuntu. Per ulteriori informazioni, consulta [Installazione degli agenti di estrazione](#).

Per configurare l'agente di estrazione dei dati, inserisci i motori di database di origine e di destinazione. Inoltre, assicurati di aver scaricato i driver JDBC per i database di origine e di destinazione sul computer in cui esegui l'agente di estrazione dati. Gli agenti di estrazione dei dati utilizzano questi driver per connettersi ai database di origine e di destinazione. Per ulteriori informazioni, consulta [Scaricamento dei driver di database richiesti](#).

In Windows, il programma di installazione dell'agente di estrazione dati avvia la procedura guidata di configurazione nella finestra del prompt dei comandi. In Linux, esegui il `sct-extractor-setup.sh` file dalla posizione in cui hai installato l'agente.

Fase 5: Configurazione AWS SCT per accedere al bucket Amazon S3

Per informazioni sulla configurazione di un bucket Amazon S3, [consulta la panoramica](#) dei bucket nella Guida per l'utente di Amazon Simple Storage Service.

Fase 6: Importa un AWS Snowball lavoro nel tuo progetto AWS SCT

Per connettere il AWS SCT progetto al dispositivo AWS Snowball Edge, importa il AWS Snowball lavoro.

Per importare il tuo AWS Snowball lavoro

1. Apri il menu Impostazioni, quindi scegli Impostazioni globali. Si aprirà la finestra di dialogo Global Settings (Impostazioni globali).
2. Scegli i profili di AWS servizio, quindi scegli Importa lavoro.
3. Scegli il tuo AWS Snowball lavoro.
4. Inserisci il tuo AWS Snowball IP. Per ulteriori informazioni, consulta [Modifica dell'indirizzo IP](#) nella Guida AWS Snowball per l'utente.

5. Inserisci la tua AWS Snowball porta. Per ulteriori informazioni, consulta [Porte necessarie per utilizzare AWS i servizi su un dispositivo AWS Snowball Edge](#) nella AWS Snowball Edge Developer Guide.
6. Inserisci la chiave di AWS Snowball accesso e la chiave AWS Snowball segreta. Per ulteriori informazioni, consulta la sezione [Autorizzazione e controllo degli accessi AWS Snowball nella Guida AWS Snowball per l'utente](#).
7. Scegli Apply (Applica), quindi OK.

Fase 7: Registrare un agente di estrazione dati in AWS SCT

In questa sezione, si registra l'agente di estrazione dati in AWS SCT.

Per registrare un agente di estrazione dati

1. Nel menu Visualizza, scegli Data migration view (altro), quindi scegli Registra.
2. Per Descrizione, inserisci un nome per l'agente di estrazione dei dati.
3. Per Nome host, inserisci l'indirizzo IP del computer su cui viene eseguito l'agente di estrazione dati.
4. Per Porta, inserisci la porta di ascolto che hai configurato.
5. Scegli Registrati.

Fase 8: Creazione di un'attività locale

Successivamente, crei l'attività di migrazione, che include due attività secondarie: Una sottoattività consente di migrare i dati dal database di origine all'appliance AWS Snowball Edge. L'altra attività secondaria acquisisce i dati caricati dall'appliance in un bucket Amazon S3 e ne esegue la migrazione nel database di destinazione.

Per creare l'attività di migrazione

1. Nel menu Visualizza, quindi scegli Data migration view (altro).
2. Nel pannello a sinistra, in cui viene visualizzato lo schema del database di origine, scegli un oggetto dello schema per il quale eseguire la migrazione. Apri il menu contestuale (fai clic con il pulsante destro del mouse) per l'oggetto, quindi scegli Crea attività locale.
3. In Nome attività, inserisci un nome descrittivo per l'attività di migrazione dei dati.
4. Per la modalità di migrazione, scegli Estrai, carica e copia.

5. Scegli le impostazioni di Amazon S3.
6. Seleziona Usa Snowball.
7. Inserisci cartelle e sottocartelle nel tuo bucket Amazon S3 in cui l'agente di estrazione dati può archiviare i dati.
8. Scegli Create (crea) per creare l'attività.

Fase 9: Esecuzione e monitoraggio dell'attività di migrazione dei dati in AWS SCT

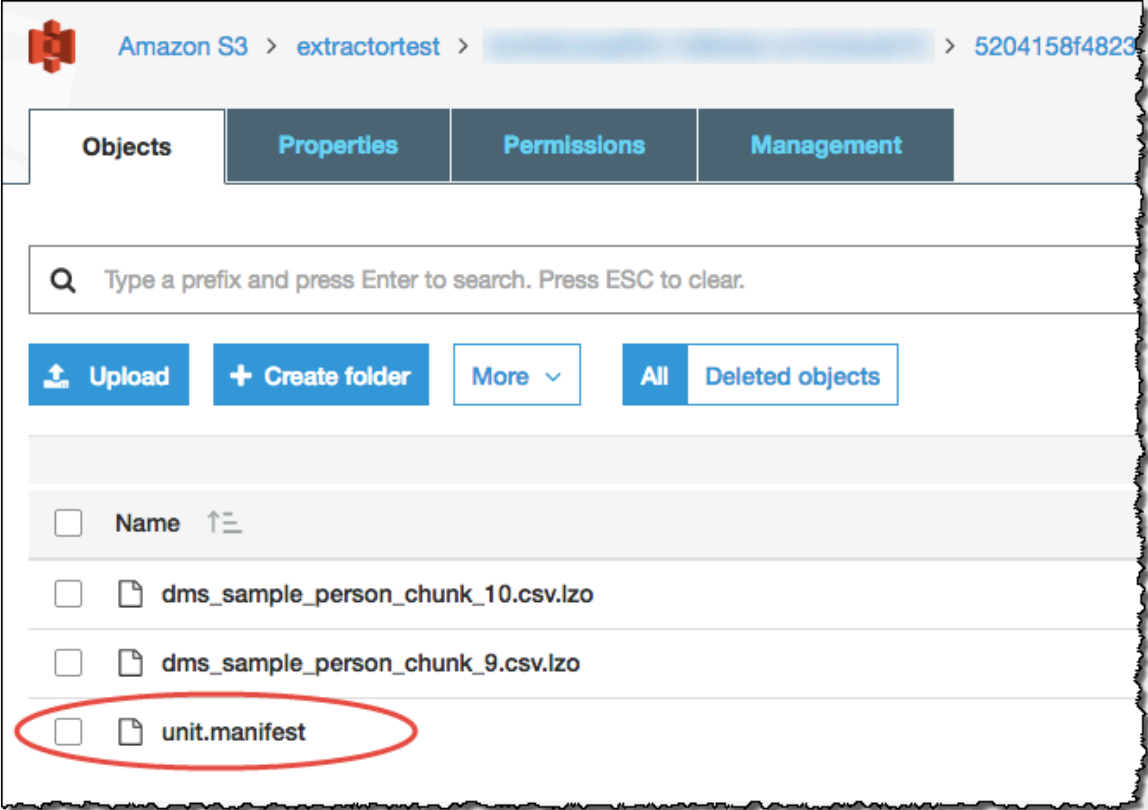
Per iniziare l'attività di migrazione dei dati, scegli Avvia. Assicurati di aver stabilito connessioni al database di origine, al bucket Amazon S3, al AWS Snowball dispositivo e alla connessione al database di destinazione su AWS.

Puoi monitorare e gestire le attività di migrazione dei dati e le relative sottoattività nella scheda Attività. È possibile visualizzare l'avanzamento della migrazione dei dati, nonché sospendere o riavviare le attività di migrazione dei dati.

Output dell'attività di estrazione dei dati

Dopo il completamento delle attività di migrazione, i tuoi dati sono pronti. Utilizza le informazioni riportate di seguito per determinare come procedere in base alla modalità di migrazione scelta e alla posizione dei dati.

Modalità di migrazione	Ubicazione dei dati
Estrai, carica e copia	I dati sono già presenti nel tuo data warehouse Amazon Redshift. Puoi verificare che i dati siano presenti e iniziare a utilizzarli. Per ulteriori informazioni, consulta Connessione ai cluster da strumenti e codice client .
Estrai e carica	<p>Gli agenti di estrazione hanno salvato i dati come file nel bucket Amazon S3. Puoi utilizzare il comando Amazon Redshift COPY per caricare i dati su Amazon Redshift. Per ulteriori informazioni, consulta Caricamento di dati da Amazon S3 nella documentazione di Amazon Redshift.</p> <p>Nel bucket Amazon S3 sono presenti più cartelle, corrispondenti alle attività di estrazione che hai impostato. Quando carichi i dati su Amazon Redshift, specifica</p>

Modalità di migrazione	Ubicazione dei dati
	<p>il nome del file manifest creato da ciascuna attività. Il file manifest viene visualizzato nella cartella delle attività del bucket Amazon S3 come illustrato di seguito.</p>  <p>The screenshot shows the Amazon S3 console interface. The breadcrumb path is 'Amazon S3 > extractortest > [redacted] > 5204158f4823'. There are tabs for 'Objects', 'Properties', 'Permissions', and 'Management'. A search bar contains the text 'Type a prefix and press Enter to search. Press ESC to clear.' Below the search bar are buttons for 'Upload', '+ Create folder', 'More', 'All', and 'Deleted objects'. A list of objects is displayed with columns for checkboxes, Name, and file icons. The objects listed are 'dms_sample_person_chunk_10.csv.lzo', 'dms_sample_person_chunk_9.csv.lzo', and 'unit.manifest'. The 'unit.manifest' file is circled in red.</p>
Estrai solo	<p>Gli agenti di estrazione hanno salvato i dati come file nella tua cartella di lavoro. Copia manualmente i dati nel bucket Amazon S3, quindi procedi con le istruzioni per l'estrazione e il caricamento.</p>

Utilizzo del partizionamento virtuale con AWS Schema Conversion Tool

Puoi gestire tabelle non partizionate di grandi dimensioni creando sottoattività che generano partizioni virtuali dei dati della tabella utilizzando le regole di filtro. In AWS SCT, puoi creare partizioni virtuali per i tuoi dati migrati. Esistono tre tipi di partizione, che funzionano con determinati tipi di dati:

- Il tipo di partizione RANGE funziona con tipi di dati numerici e relativi a data e ora.

- Il tipo di partizione LIST funziona con tipi di dati numerici, di carattere e relativi a data e ora.
- Il tipo di partizione DATE AUTO SPLIT funziona con tipi di dati numerici, di data e ora.

AWS SCT convalida i valori forniti per la creazione di una partizione. Ad esempio, se si tenta di partizionare una colonna con il tipo di dati NUMERIC ma si forniscono valori di un tipo di dati diverso, AWS SCT genera un errore.

Inoltre, se utilizzi AWS SCT per migrare i dati su Amazon Redshift, puoi utilizzare il partizionamento nativo per gestire la migrazione di tabelle di grandi dimensioni. Per ulteriori informazioni, consulta [Utilizzo del partizionamento nativo](#).

Limiti durante la creazione di partizionamenti virtuali

Queste sono limitazioni per la creazione di una partizione virtuale:

- Puoi utilizzare il partizionamento virtuale solo per le tabelle non partizionate.
- Puoi usare il partizionamento virtuale solo nella vista della migrazione dei dati.
- Non puoi utilizzare l'opzione UNION ALL VIEW con il partizionamento virtuale.

Tipo di partizione RANGE

Il tipo di partizione RANGE suddivide i dati in base a una gamma di valori di colonna per i tipi di dati numerici e relativi a data e ora. Questo tipo di partizione crea una clausola WHERE e tu fornisci l'intervallo di valori per ogni partizione. Per specificare un elenco di valori per la colonna partizionata, utilizza la casella Valori. Puoi caricare le informazioni sul valore utilizzando un file csv.

Il tipo di partizione RANGE crea partizioni predefinite a entrambe le estremità dei valori della partizione. Queste partizioni predefinite catturano tutti i dati inferiori o superiori ai valori di partizione specificati.

Ad esempio, è possibile creare più partizioni in base a un intervallo di valori fornito dall'utente. In questo esempio, i valori del partizionamento LO_TAX sono specificati per creare più partizioni.

```
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
```


Per creare una partizione virtuale RANGE

1. Aprire AWS SCT.
2. Scegli la modalità di visualizzazione Data Migration (altro).
3. Scegliere la tabella in cui desideri impostare il partizionamento virtuale. Apri il menu contestuale (fai clic con il pulsante destro del mouse) per la tabella e scegli Aggiungi partizionamento virtuale.
4. Nella finestra di dialogo Aggiungi partizionamento virtuale, immettete le informazioni come segue.

Opzione	Azione
Tipo di partizione	Scegli RANGE. L'interfaccia utente della finestra di dialogo cambia a seconda del tipo scelto.
Nome della colonna	Scegli la colonna su cui desideri effettuare la partizione.
Tipo di colonna	Scegli il tipo di dati per i valori nella colonna.
Valori	Aggiungere nuovi valori digitando ogni valore nella casella New Value (Nuovo valore), quindi scegliendo il segno più per aggiungere il valore.
Carica da file	(Facoltativo) Immetti il nome di un file.csv che contiene i valori della partizione.

5. Scegli OK.

Tipo di partizione LIST

Il tipo di partizione LIST suddivide i dati in base ai valori di colonna per i tipi di dati numerici, di carattere e relativi a data e ora. Questo tipo di partizione crea una clausola WHERE e tu fornisci i valori per ogni partizione. Per specificare un elenco di valori per la colonna partizionata, utilizzate la casella Valori. Puoi caricare le informazioni sul valore utilizzando un file csv.

Ad esempio, puoi creare più partizioni in base a un valore fornito. In questo esempio, i valori del partizionamento LO_ORDERKEY sono specificati per creare più partizioni.

```

Partition1: WHERE LO_ORDERKEY = 1
Partition2: WHERE LO_ORDERKEY = 2
Partition3: WHERE LO_ORDERKEY = 3
...
PartitionN: WHERE LO_ORDERKEY = USER_VALUE_N

```

Puoi anche creare una partizione di default per i valori non inclusi in quelli specificati.

È possibile utilizzare il tipo di partizione LIST per filtrare i dati di origine se si desidera escludere valori particolari dalla migrazione. Ad esempio, supponiamo di voler omettere le righe con `LO_ORDERKEY = 4`. In questo caso, non includete il valore 4 nell'elenco dei valori di partizione e assicuratevi che l'opzione `Includi altri valori` non sia selezionata.

Per creare una partizione virtuale LIST

1. Apri AWS SCT.
2. Scegli la modalità di visualizzazione Data Migration (altro).
3. Scegliere la tabella in cui desideri impostare il partizionamento virtuale. Apri il menu contestuale (fai clic con il pulsante destro del mouse) per la tabella e scegli `Aggiungi partizionamento virtuale`.
4. Nella finestra di dialogo `Aggiungi partizionamento virtuale`, immettete le informazioni come segue.

Opzione	Azione
Tipo di partizione	Scegli LIST. L'interfaccia utente della finestra di dialogo cambia a seconda del tipo scelto.
Nome della colonna	Scegli la colonna su cui desideri effettuare la partizione.
Nuovo valore	Digita un valore qui per aggiungerlo al set di valori di partizionamento.
Includi altri valori	Scegli questa opzione per creare una partizione di default in cui tutti i valori che non soddisfano i criteri di partizionamento vengono archiviati.
Carica da file	(Facoltativo) Immetti il nome di un file.csv che contiene i valori della partizione.

5. Scegli OK.

Tipo di partizione DATE AUTO SPLIT

Il tipo di partizione DATE AUTO SPLIT è un modo automatico per generare partizioni RANGE. Con DATA AUTO SPLIT, si indica AWS SCT all'attributo di partizionamento, dove iniziare e finire e la dimensione dell'intervallo tra i valori. Quindi AWS SCT calcola automaticamente i valori delle partizioni.

DATA AUTO SPLIT automatizza gran parte del lavoro relativo alla creazione di partizioni di intervallo. Il compromesso tra l'utilizzo di questa tecnica e il partizionamento a intervalli è il controllo necessario sui limiti delle partizioni. Il processo di suddivisione automatica crea sempre intervalli di dimensioni uguali (uniformi). Il partizionamento degli intervalli consente di variare le dimensioni di ciascun intervallo in base alle esigenze di una particolare distribuzione dei dati. Ad esempio, è possibile utilizzare giornalmente, settimanalmente, bisettimanalmente, mensilmente e così via.

```
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'
```

Per creare una partizione virtuale DATE AUTO SPLIT

1. Aperta AWS SCT.
2. Scegli la modalità di visualizzazione Data Migration (altro).
3. Scegliere la tabella in cui desideri impostare il partizionamento virtuale. Apri il menu contestuale (fai clic con il pulsante destro del mouse) per la tabella e scegli Aggiungi partizionamento virtuale.
4. Nella finestra di dialogo Aggiungi partizionamento virtuale, immettete le informazioni come segue.

Opzione	Azione
Tipo di partizione	Scegliere DATE AUTO SPLIT. L'interfaccia utente della finestra di dialogo cambia a seconda del tipo scelto.

Opzione	Azione
Nome della colonna	Scegli la colonna su cui desideri effettuare la partizione.
Data di inizio	Digitare una data di inizio.
Data di fine	Digitare una data di fine.
Interval (Intervallo)	Immetti l'unità dell'intervallo e scegli il valore per quell'unità.

5. Scegli OK.

Utilizzo del partizionamento nativo

Per accelerare la migrazione dei dati, gli agenti di estrazione dei dati possono utilizzare partizioni native di tabelle sul server del data warehouse di origine. AWS SCT supporta il partizionamento nativo per le migrazioni da Greenplum, Netezza e Oracle ad Amazon Redshift.

Ad esempio, dopo aver creato un progetto, è possibile raccogliere statistiche su uno schema e analizzare le dimensioni delle tabelle selezionate per la migrazione. Per le tabelle che superano la dimensione specificata, AWS SCT attiva il meccanismo di partizionamento nativo.

Per utilizzare il partizionamento nativo

1. Apri AWS SCT e scegli Nuovo progetto per File. Viene visualizzata la finestra di dialogo Nuovo progetto.
2. Crea un nuovo progetto, aggiungi i server di origine e di destinazione e crea regole di mappatura. Per ulteriori informazioni, consulta [Creare unAWS SCT progetto](#).
3. Scegliete Visualizza, quindi scegliete Visualizzazione principale.
4. Per le impostazioni del progetto, scegli la scheda Migrazione dei dati. Scegli Usa il partizionamento automatico. Per i database di origine Greenplum e Netezza, inserisci la dimensione minima delle tabelle supportate in megabyte (ad esempio, 100). AWS SCT crea automaticamente sottoattività di migrazione separate per ogni partizione nativa che non sia vuota. Per le migrazioni da Oracle ad Amazon Redshift, AWS SCT crea sottoattività per tutte le tabelle partizionate.
5. Nel pannello di sinistra che mostra lo schema dal database di origine, scegli uno schema. Apri il menu contestuale (fai clic con il pulsante destro del mouse) per l'oggetto e scegli Raccogli

statistiche. Per la migrazione dei dati da Oracle ad Amazon Redshift, puoi saltare questo passaggio.

6. Scegli tutte le tabelle da migrare.
7. Registra il numero richiesto di agenti. Per ulteriori informazioni, consulta [Registrazione degli agenti di estrazione con AWS Schema Conversion Tool](#).
8. Crea un'attività di estrazione dei dati per le tabelle selezionate. Per ulteriori informazioni, consulta [Creazione, esecuzione e monitoraggio di un'attività di estrazione dei dati AWS SCT](#).

Controlla se le tabelle di grandi dimensioni sono suddivise in sottoattività e che ciascuna sottoattività corrisponda al set di dati che presenta una parte della tabella situata su una sezione del data warehouse di origine.

9. Avvia e monitora il processo di migrazione fino a quando gli agenti di estrazione AWS SCT dei dati non completano la migrazione dei dati dalle tabelle di origine.

Migrazione dei LOB su Amazon Redshift

Amazon Redshift non supporta lo storage di oggetti binari di grandi dimensioni (LOB). Tuttavia, se devi migrare uno o più LOB su Amazon Redshift AWS SCT, puoi eseguire la migrazione. A tale scopo, AWS SCT utilizza un bucket Amazon S3 per archiviare i LOB e scrive l'URL del bucket Amazon S3 nei dati migrati archiviati in Amazon Redshift.

Per migrare i LOB su Amazon Redshift

1. Aprire un progetto. AWS SCT
2. Esegui la connessione ai database di origine e di destinazione. Aggiorna i metadati dal database di destinazione e assicurati che le tabelle convertite esistano lì.
3. Per Azioni, scegli Crea attività locale.
4. Per la modalità di migrazione, scegli una delle seguenti opzioni:
 - Estrai e carica per estrarre i tuoi dati e carica i tuoi dati su Amazon S3.
 - Estrai, carica e copia per estrarre i tuoi dati, carica i dati su Amazon S3 e copiali nel tuo data warehouse Amazon Redshift.
5. Scegli le impostazioni di Amazon S3.
6. Per la cartella LOB del bucket Amazon S3, inserisci il nome della cartella in un bucket Amazon S3 in cui desideri archiviare i LOB.

Se utilizzi il profilo di AWS servizio, questo campo è facoltativo. AWS SCT puoi utilizzare le impostazioni predefinite del tuo profilo. Per utilizzare un altro bucket Amazon S3, inserisci il percorso qui.

7. Attiva l'opzione Usa proxy per utilizzare un server proxy per caricare dati su Amazon S3. Quindi scegli il protocollo di trasferimento dati, inserisci il nome host, la porta, il nome utente e la password.
8. Per il tipo di endpoint, scegli FIPS per utilizzare l'endpoint FIPS (Federal Information Processing Standard). Scegli VPCE per utilizzare l'endpoint del cloud privato virtuale (VPC). Quindi, per l'endpoint VPC, inserisci il Domain Name System (DNS) del tuo endpoint VPC.
9. Attiva l'opzione Keep files on Amazon S3 dopo averli copiati su Amazon Redshift per conservare i file estratti su Amazon S3 dopo averli copiati su Amazon Redshift.
10. Scegli Create (crea) per creare l'attività.

Migliori pratiche e risoluzione dei problemi per gli agenti di estrazione dei dati

Di seguito sono elencati alcuni suggerimenti per la risoluzione dei problemi e best practice per l'utilizzo degli agenti di estrazione.

Problema	Suggerimenti sulla risoluzione dei problemi
Le prestazioni sono lente	<p>Per migliorare le prestazioni, ti consigliamo di attenerti alle seguenti indicazioni:</p> <ul style="list-style-type: none"> • Installa più agenti. • Installa agenti su computer in prossimità del data warehouse. • Non eseguire tutte le tabelle su una sola attività dell'agente.
Ritardi dovuti a conflitti	Evita la presenza di troppi agenti che accedono al tuo data warehouse nello stesso momento.
Un agente non è temporaneamente disponibile	Se un agente non è disponibile, lo stato di ogni sua attività viene visualizzato come non riuscito in AWS SCT. Se attendi, in alcuni

Problema	Suggerimenti sulla risoluzione dei problemi
	<p>casi l'agente è in grado di tornare disponibile. In questo caso, lo stato delle sue attività si aggiorna in AWS SCT.</p>
Un agente non è disponibile in modo permanente	<p>Se il computer che esegue un agente diventa non disponibile in modo permanente e tale agente sta eseguendo un'attività, puoi sostituirlo con un nuovo agente per continuare l'attività. Puoi sostituire un nuovo agente solo se la cartella di lavoro dell'agente originale non era nello stesso computer dell'agente originale. Per sostituire un nuovo agente, esegui le operazioni indicate di seguito:</p> <ul style="list-style-type: none">• Installa un agente su un nuovo computer.• Configura il nuovo agente con le stesse impostazioni, compresi il numero di porta e la cartella di lavoro, come fosse l'agente originale.• Avviare l'agente. Dopo che l'agente viene avviato, l'attività individua il nuovo agente disponibile e continua l'esecuzione sul nuovo agente.

Conversione del codice SQL dell'applicazione utilizzando AWS SCT

Quando converti lo schema del database da un motore a un altro, è anche necessario aggiornare il codice SQL nelle applicazioni, per interagire con il nuovo motore di database al posto di quello precedente. Puoi visualizzare, analizzare, modificare e salvare il codice SQL convertito.

Puoi utilizzare AWS Schema Conversion Tool (AWS SCT) per convertire il codice SQL in codice C++, C#, Java o altro tipo di codice dell'applicazione. Per una conversione da Oracle a PostgreSQL, puoi usare AWS SCT per convertire il codice SQL*Plus in PSQL. Inoltre, per una conversione da Oracle a PostgreSQL, puoi usare AWS SCT per convertire il codice SQL incorporato nelle applicazioni C#, C++, Java e Pro*C.

Argomenti

- [Panoramica sulla conversione dell'applicazione SQL](#)
- [Conversione del codice SQL nelle applicazioni con AWS SCT](#)
- [Conversione del codice SQL in applicazioni C# con AWS SCT](#)
- [Conversione del codice SQL in applicazioni C++ con AWS SCT](#)
- [Conversione del codice SQL in applicazioni Java con AWS SCT](#)
- [Conversione del codice SQL nelle applicazioni Pro*C con AWS SCT](#)

Panoramica sulla conversione dell'applicazione SQL

Per convertire il codice SQL nella tua applicazione, esegui i seguenti passaggi di alto livello:

- Crea un progetto di conversione dell'applicazione— Il progetto di conversione dell'applicazione è un figlio del progetto di conversione dello schema del database. Ogni progetto di conversione dello schema del database può avere uno o più progetti di conversione dell'applicazione secondari. Per ulteriori informazioni, consulta [Creazione di progetti generici di conversione di applicazioni in AWS SCT](#).
- Analizza e converti il tuo codice SQL—AWS SCT analizza l'applicazione, estrae il codice SQL e crea una versione locale dell'SQL convertito da rivedere e modificare. Lo strumento non modifica il codice della tua applicazione fino a quando non sei pronto. Per ulteriori informazioni, consulta [Analisi e conversione del codice SQL in AWS SCT](#).

- Crea un rapporto di valutazione dell'applicazione— Il rapporto di valutazione dell'applicazione fornisce informazioni importanti sulla conversione del codice SQL dell'applicazione dallo schema del database di origine allo schema del database di destinazione. Per ulteriori informazioni, consulta [Creazione e utilizzo diAWS SCTrapporto di valutazione inAWS SCT](#).
- Modifica, applica modifiche e salva il codice SQL convertito— Il rapporto di valutazione include un elenco di elementi del codice SQL che non possono essere convertiti automaticamente. Per questi item, puoi modificare il codice SQL manualmente per eseguire la conversione. Per ulteriori informazioni, consulta [Modifica e salvataggio del codice SQL convertito conAWS SCT](#).

Conversione del codice SQL nelle applicazioni conAWS SCT

Puoi usareAWS SCTper convertire il codice SQL incorporato nelle tue applicazioni. Il genericoAWS SCTApplication Converter tratta il codice dell'applicazione come testo semplice. Scansiona il codice dell'applicazione ed estrae il codice SQL con espressioni regolari. Questo convertitore supporta diversi tipi di file di codice sorgente e funziona con codice applicativo scritto in qualsiasi linguaggio di programmazione.

Il convertitore di applicazioni generico presenta le seguenti limitazioni. Non approfondisce la logica dell'applicazione specifica per il linguaggio di programmazione dell'applicazione. Inoltre, il convertitore generico non supporta istruzioni SQL provenienti da diversi oggetti dell'applicazione, come funzioni, parametri, variabili locali e così via.

Per migliorare la conversione del codice SQL dell'applicazione, utilizzate convertitori di codice SQL applicativi specifici della lingua. Per ulteriori informazioni, consulta [Conversione del codice SQL in applicazioni C#](#), [Conversione del codice SQL nelle applicazioni Java](#) e [Conversione del codice SQL nelle applicazioni Pro*C](#).

Creazione di progetti generici di conversione di applicazioni inAWS SCT

In AWS Schema Conversion Tool, il progetto di conversione dell'applicazione è secondario al progetto di conversione dello schema del database. Ogni progetto di conversione dello schema del database può avere uno o più progetti di conversione dell'applicazione secondari.

Note

AWS SCTnon supporta la conversione tra le seguenti fonti e destinazioni:

- Oracle a Oracle

- Da PostgreSQL a PostgreSQL o Aurora PostgreSQL
- MySQL a MySQL
- da SQL Server a SQL Server
- Da Amazon Redshift ad Amazon Redshift
- Da SQL Server a Babelfish
- Servizi di integrazione con SQL Server per AWS Glue
- Da Apache Cassandra ad Amazon DynamoDB

Utilizzare la procedura seguente per creare un progetto di conversione di applicazioni generico.

Creazione di un progetto di conversione dell'applicazione

1. Nel **AWS Schema Conversion Tool**, scegli **Nuova applicazione generica** sul menu **Applicazioni**.

Verrà visualizzata la finestra di dialogo **New application conversion project** (Nuovo progetto di conversione dell'applicazione).

Creating a generic application conversion project

Enter the name, location and type of the new application conversion project.

Name: Application conversion project 1

Location: C:\AWS-SCT-Demo Browse

Language: Java Target parameter style: Same as in source

Settings

Don't cast bind variables to SQL types i

Keep object names i

Choose the source database schema that your application uses which is mapped with the target tree object:

- ▼ Schemas [58]
 - ANONYMOUS
 - APPQOSSYS
 - AUDSYS
 - CHINOOK**
 - CTXSYS
 - DVSY

OK Cancel

2. Aggiungi le informazioni del progetto seguenti.

Per questo parametro	Eeguire questa operazione
Nome	Immettere un nome per il progetto di conversione dell'applicazione. Ogni progetto di conversione dello schema del database può avere uno o più progetti di conversione dell'applicazione secondari, quindi scegli un nome significativo se aggiungi più progetti in seguito.
Location (Ubicazione)	Immetti l'ubicazione del codice sorgente per la tua applicazione.
Linguaggio	Scegli una delle seguenti opzioni: <ul style="list-style-type: none"> • Java

Per questo parametro	Eeguire questa operazione
	<ul style="list-style-type: none"> • C++ • C# • Any (Qualsiasi)
stile del parametro di destinazione	<p>Scegliete la sintassi da usare per le variabili di associazione nel codice convertito. Diverse piattaforme di database utilizzano una sintassi diversa per le variabili di associazione. Seleziona una delle seguenti opzioni:</p> <ul style="list-style-type: none"> • Same as in source (Come in sorgente) • Positional (?) (Posizionale (?)) • Indexed (:1) (Indicizzato (:1)) • Indexed (\$1) (Indicizzato (\$1)) • Named (@name) (Denominato (@nome)) • Named (:name) (Denominato (:nome)) • Denominato (&nome) • Denominato (\$name) • Denominato (#name) • Denominato (! nome!)
Scegliere lo schema del database di origine	Nell'albero dei sorgenti, scegli lo schema utilizzato dall'applicazione. Assicurati che questo schema faccia parte di una regola di mappatura.

3. `SelezioneNon` trasmette le variabili di associazione ai tipi SQL per evitare la conversione dei tipi di variabili di associazione in tipi SQL. Questa opzione è disponibile solo per la conversione da Oracle a PostgreSQL.

Ad esempio, il codice dell'applicazione sorgente include la seguente query Oracle:

```
SELECT * FROM ACCOUNT WHERE id = ?
```

Quando si seleziona `Non trasferite variabili di associazione a tipi SQL`, AWS SCT converte questa richiesta come illustrato di seguito.

```
SELECT * FROM account WHERE id = ?
```

Quando si cancellaNon trasferite variabili di associazione a tipi SQL,AWS SCTcambia il tipo di variabile di associazione inNUMERICtipo di dati Il risultato della conversione è illustrato di seguito.

```
SELECT * FROM account WHERE id = (?)::NUMERIC
```

4. SelezionaMantieni i nomi degli oggetti per evitare di aggiungere il nome dello schema al nome dell'oggetto convertito. Questa opzione è disponibile solo per la conversione da Oracle a PostgreSQL.

Ad esempio, supponiamo che il codice dell'applicazione sorgente includa la seguente query Oracle.

```
SELECT * FROM ACCOUNT
```

Quando si selezionaMantieni i nomi degli oggetti,AWS SCTconverte questa richiesta come illustrato di seguito.

```
SELECT * FROM account
```

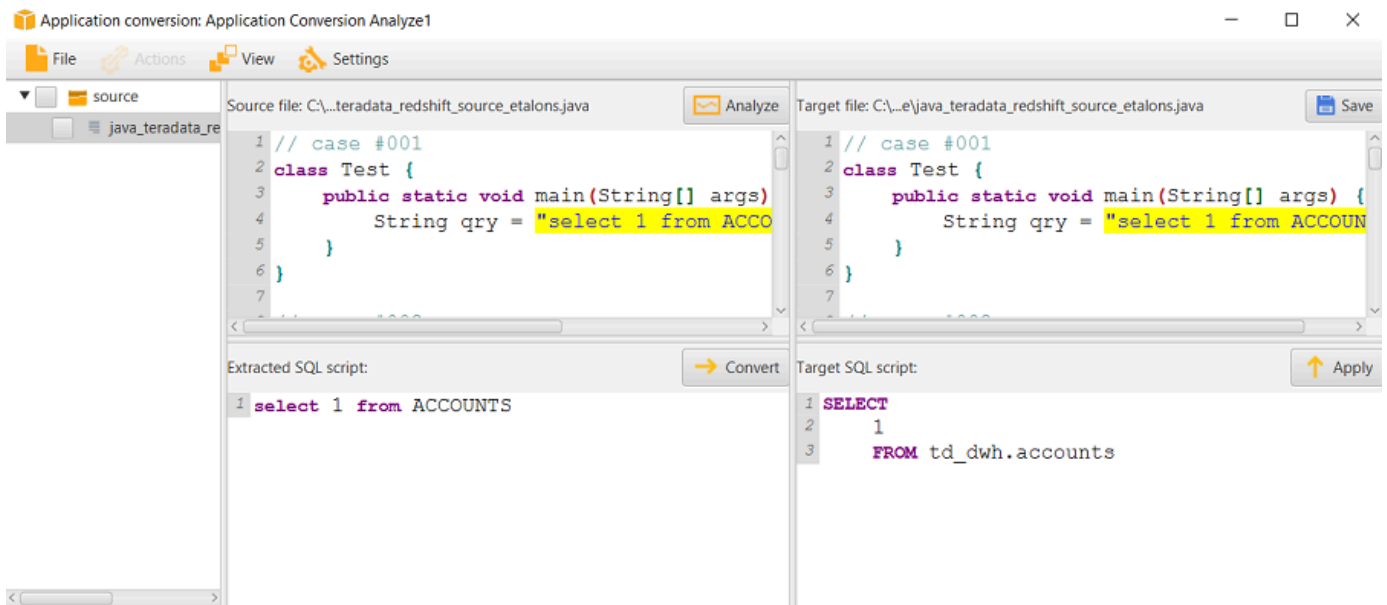
Quando si cancellaMantieni i nomi degli oggetti,AWS SCTaggiunge il nome dello schema al nome della tabella. Il risultato della conversione è illustrato di seguito.

```
SELECT * FROM schema_name.account
```

Se il codice sorgente include i nomi degli oggetti principali nei nomi degli oggetti,AWS SCTutilizza questo formato nel codice convertito. In questo caso, ignora ilMantieni i nomi degli oggettiopzione perchéAWS SCTaggiunge i nomi degli oggetti principali nel codice convertito.

5. Scegli OK per creare il tuo progetto di conversione dell'applicazione.

Si aprirà la finestra del progetto.



Gestione dei progetti di conversione delle applicazioni in AWS SCT

È possibile aprire un progetto di conversione di applicazioni esistente e aggiungere più progetti di conversione di applicazioni.

Dopo aver creato un progetto di conversione di applicazioni, la finestra del progetto si apre automaticamente. È possibile chiudere la finestra del progetto di conversione dell'applicazione e riprenderla in un secondo momento.

Per aprire un progetto di conversione di applicazioni esistente

1. Nel pannello di sinistra (a sinistra), scegli il nodo del progetto di conversione dell'applicazione e apri il menu contestuale (fai clic con il pulsante destro del mouse).
2. Scegliete Gestisci l'applicazione.

Per aggiungere un ulteriore progetto di conversione delle applicazioni

1. Nel pannello di sinistra (a sinistra), scegli il nodo del progetto di conversione dell'applicazione e apri il menu contestuale (fai clic con il pulsante destro del mouse).
2. Scegli New application (Nuova applicazione).
3. Inserire le informazioni necessarie per creare un nuovo progetto di conversione dell'applicazione. Per ulteriori informazioni, consulta [Creazione di progetti generici di conversione di applicazioni](#).

Analisi e conversione del codice SQL inAWS SCT

Utilizzare la procedura seguente per analizzare e convertire il codice SQL inAWS Schema Conversion Tool.

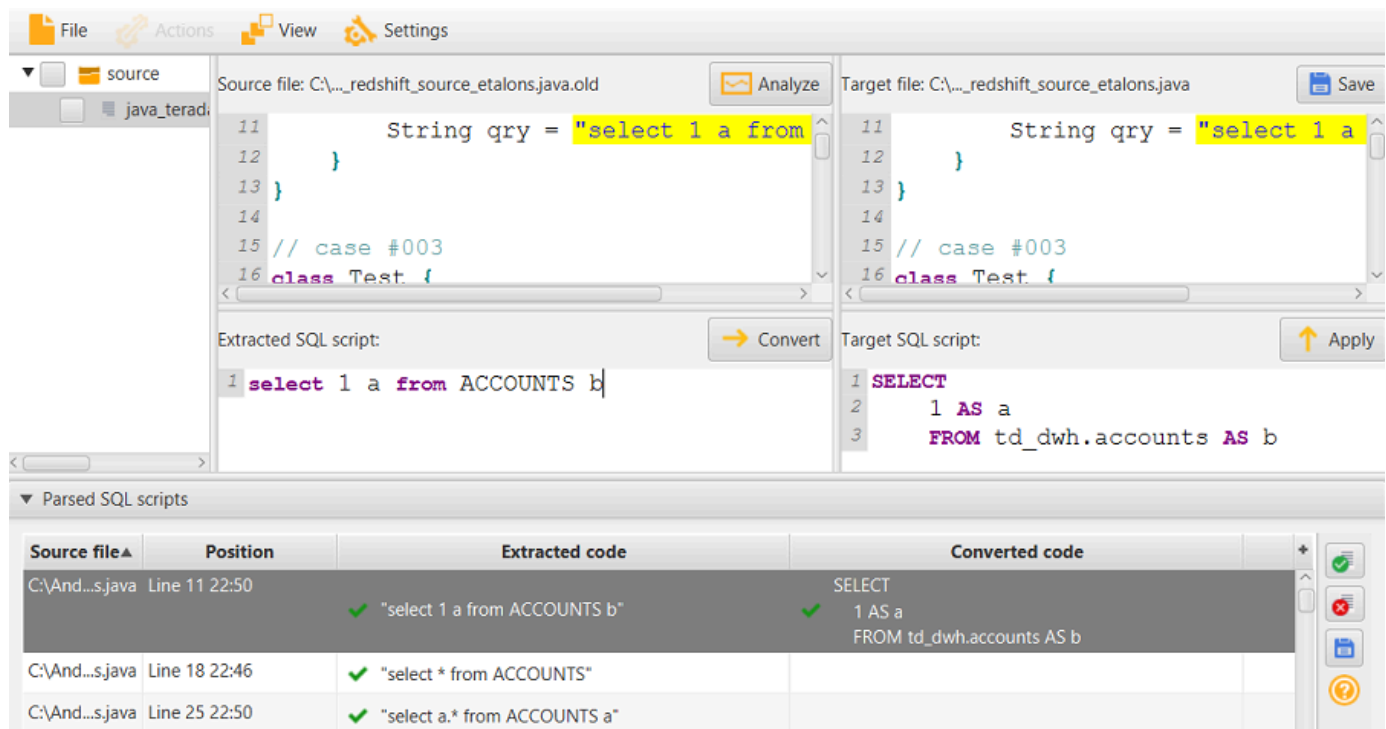
Analisi e Conversione del codice SQL

1. Apri un progetto di conversione di applicazioni esistente e scegli **Analizza**.

AWS SCT analizza il codice dell'applicazione ed estrae il codice SQL. AWS SCT visualizza il codice SQL estratto nel script SQL analizzato elenco.

2. Per Script SQL analizzati, scegli un elemento per esaminarne il codice SQL estratto. AWS SCT visualizza il codice dell'elemento selezionato nel script SQL estratto lastra.
3. Scegli **Convertire** per convertire il codice SQL in script SQL estratto lastra. AWS SCT converte il codice in un formato compatibile con il database di destinazione.

È possibile modificare il codice SQL convertito. Per ulteriori informazioni, consulta [Modifica e salvataggio del codice SQL convertito](#).



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. Quando crei un rapporto di valutazione della conversione delle applicazioni, AWS SCT converte tutti gli elementi del codice SQL estratti. Per ulteriori informazioni, consulta [Creazione e utilizzo del rapporto di valutazione](#).

Creazione e utilizzo diAWS SCTrapporto di valutazione inAWS SCT

Il rapporto di valutazione della conversione delle applicazionifornisce informazioni sulla conversione del codice SQL dell'applicazione in un formato compatibile con il database di destinazione. Il rapporto descrive in dettaglio tutto il codice SQL estratto, tutto il codice SQL convertito e le azioni per il codice SQL cheAWS SCTimpossibile convertire.

Creazione di un rapporto di valutazione della conversione delle applicazioni

Utilizzare la procedura seguente per creare un rapporto di valutazione della conversione delle applicazioni.

Per creare un rapporto di valutazione della conversione delle applicazioni

1. Nella finestra del progetto di conversione dell'applicazione, scegliaccedi al databasesulAzionimenu.

AWS SCTcrea il rapporto di valutazione della conversione dell'applicazione e lo apre nella finestra del progetto di conversione dell'applicazione.

2. Verifica la scheda Summary (Riepilogo).

IlRiepilogola scheda, mostrata di seguito, mostra le informazioni di riepilogo del rapporto di valutazione dell'applicazione. Essa mostra gli item del codice SQL che sono stati convertiti automaticamente e quelli che non sono stati automaticamente.



3. ScegliAzioni di estrazione SQL.

Esamina l'elenco degli elementi del codice SQL cheAWS SCTnon puoi estrarre dal tuo codice sorgente.

4. Sceglieteazioni di conversione SQL.

Esamina l'elenco degli elementi del codice SQL cheAWS SCTnon può essere convertito automaticamente. Utilizza le azioni consigliate per convertire manualmente il codice SQL. Per informazioni su come modificare il codice SQL convertito, vedere[Modifica e salvataggio del codice SQL convertito conAWS SCT](#).

5. (Facoltativo) Salvare una copia locale del rapporto come file PDF o come file con valori separati da virgole (CSV):

- Scegli **Salvataggio del PDF** in alto a destra per salvare il rapporto come file PDF.

Il file PDF contiene il riepilogo esecutivo, le azioni da intraprendere e i consigli per la conversione delle applicazioni.

- Scegli **teaccedi al file CSV** in alto a destra per salvare il rapporto come file CSV.

Il file CSV contiene azioni, azioni consigliate e una stima della complessità dello sforzo manuale necessario per convertire il codice SQL.

Modifica e salvataggio del codice SQL convertito con AWS SCT

Il rapporto di valutazione include un elenco di elementi del codice SQL che AWS SCT impossibile convertire. Per ogni articolo, AWS SCT crea un elemento d'azione su azioni di conversione SQL scheda. Per questi item, puoi modificare il codice SQL manualmente per eseguire la conversione.

Utilizza la procedura seguente per modificare il tuo codice SQL convertito, per applicare le modifiche e quindi salvarle.

Modifica, applicazione delle modifiche e salvataggio del codice SQL convertito

1. Modifica il tuo codice SQL convertito direttamente nel riquadro Target SQL script (Script SQL di destinazione). Se non viene mostrato alcun codice convertito, puoi fare clic nel riquadro e iniziare a digitare;
2. Dopo aver modificato il codice SQL convertito, scegli **Apply (Applica)**. A questo punto, le modifiche vengono salvate in memoria, ma non ancora scritte nel tuo file;
3. Scegli **Save (Salva)** per salvare le modifiche nel tuo file.

Quando scegli **Salva**, sovrascrivi il file originale. Crea una copia del file originale prima di salvare, in modo da disporre di un record del tuo codice dell'applicazione originale.

Conversione del codice SQL in applicazioni C# con AWS SCT

Per una conversione da Oracle a PostgreSQL, puoi usare AWS Schema Conversion Tool (AWS SCT) per convertire il codice SQL incorporato nelle applicazioni C#. Questo convertitore di applicazioni C#

specifico comprende la logica dell'applicazione. Raccoglie istruzioni che si trovano in diversi oggetti dell'applicazione, come funzioni, parametri, variabili locali e così via.

Grazie a questa analisi approfondita, il convertitore di codice SQL dell'applicazione C# offre risultati di conversione migliori rispetto al convertitore generico.

Creazione di progetti di conversione di applicazioni C# in AWS SCT

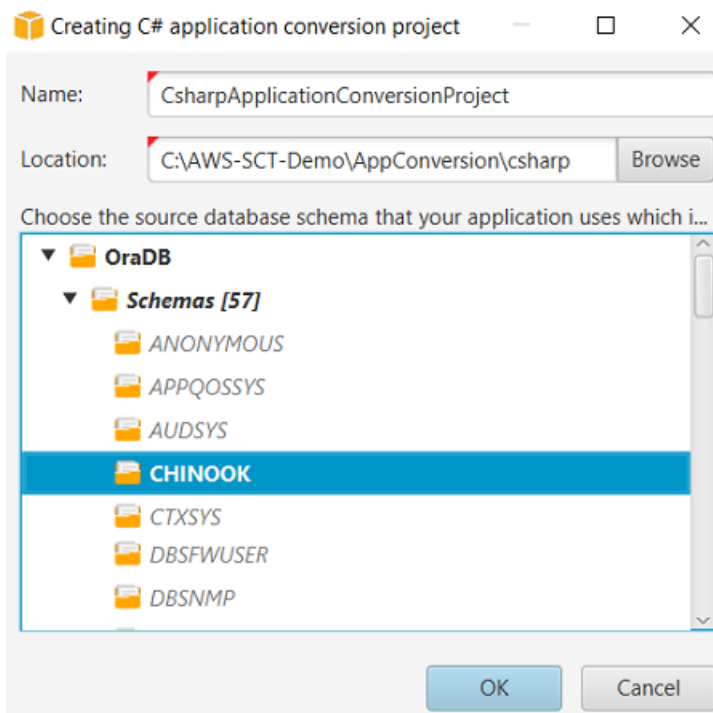
È possibile creare un progetto di conversione di applicazioni C# solo per convertire gli schemi di database Oracle in schemi di database PostgreSQL. Assicurati di aggiungere una regola di mappatura nel tuo progetto che includa uno schema Oracle di origine e un database PostgreSQL di destinazione. Per ulteriori informazioni, consulta [Creazione di regole di mappatura in AWS SCT](#).

È possibile aggiungere più progetti di conversione di applicazioni in un unico AWS SCT progetto. Utilizzare la procedura seguente per creare un progetto di conversione di applicazioni C#.

Per creare un progetto di conversione di applicazioni C#

1. Creare un progetto di conversione del database e aggiungere un database Oracle di origine. Per ulteriori informazioni, consultare [Creare un AWS SCT progetto](#) e [Aggiungere server di database a un AWS SCT progetto](#).
2. Aggiungere una regola di mappatura che includa il tuo database Oracle di origine e un database PostgreSQL di destinazione. È possibile aggiungere un database PostgreSQL di destinazione o utilizzare una piattaforma di database di destinazione PostgreSQL virtuale in una regola di mappatura. Per ulteriori informazioni, consultare [Creazione di regole di mappatura in AWS SCT](#) e [Utilizzo di obiettivi virtuali](#).
3. Sul Visualizza menu, scegliere Vista principale.
4. Sul Applicazioni menu, scegliere Nuova applicazione C#.

La creazione di un progetto di conversione di applicazioni C# viene visualizzata una finestra di dialogo.



5. PerNome, inserire un nome per il progetto di conversione dell'applicazione C#. Poiché ogni progetto di conversione dello schema del database può avere uno o più progetti di conversione di applicazioni secondarie, scegliete un nome che abbia senso se aggiungete più progetti.
6. PerPosizione, inserire la posizione del codice sorgente dell'applicazione.
7. Nell'albero dei sorgenti, scegli lo schema utilizzato dall'applicazione. Assicurati che questo schema faccia parte di una regola di mappatura. AWS SCT evidenzia in grassetto gli schemi che fanno parte di una regola di mappatura.
8. Scegliete OK per creare il tuo progetto di conversione dell'applicazione C#.
9. Trova il tuo progetto di conversione di applicazioni C# nel Applicazione nodo nel pannello di sinistra.

Conversione del codice SQL dell'applicazione C# in AWS SCT

Dopo aver aggiunto l'applicazione C# al AWS SCT project, converti il codice SQL di questa applicazione in un formato compatibile con la piattaforma di database di destinazione. Utilizzare la procedura seguente per analizzare e convertire il codice SQL incorporato nell'applicazione C# in AWS Schema Conversion Tool.

Per convertire il codice SQL

1. Espandi il C# nodo sotto Applicazioni nel pannello di sinistra.
2. Scegliere l'applicazione da convertire e aprire il menu contestuale (pulsante destro del mouse).
3. Scegliete Convertire. AWS SCT analizza i file del codice sorgente, determina la logica dell'applicazione e carica i metadati del codice nel progetto. Questi metadati di codice includono classi C#, oggetti, metodi, variabili globali, interfacce e così via.

nel pannello del database di destinazione, AWS SCT crea una struttura di cartelle simile a quella del progetto dell'applicazione di origine. Qui puoi rivedere il codice dell'applicazione convertito.

```
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. Salvare il codice dell'applicazione convertito. Per ulteriori informazioni, consulta [Salvare il codice dell'applicazione convertito](#).

Le tue applicazioni C# potrebbero includere codice SQL che interagisce con diversi database di origine. È possibile migrare a PostgreSQL diversi di questi database di origine. In questo caso, assicurati di non convertire il codice SQL che interagisce con i database che hai escluso dall'ambito della migrazione. È possibile escludere i file di origine dell'applicazione C# dall'ambito di conversione. A tale scopo, deselezionate le caselle di controllo relative ai nomi dei file che desiderate escludere dall'ambito di conversione.

Dopo aver modificato l'ambito di conversione, AWS SCT analizza comunque il codice SQL di tutti i file sorgente delle applicazioni C#. Quindi, AWS SCT copia nella cartella di destinazione tutti i file di origine esclusi dall'ambito di conversione. Questa operazione consente di creare l'applicazione dopo aver salvato i file dell'applicazione convertiti.

Salvare il codice dell'applicazione convertito con AWS SCT

Utilizzare la procedura seguente per salvare il codice dell'applicazione convertito.

Per salvare il codice dell'applicazione convertito

1. Espandi il C# nodo sotto Applicazioni nel pannello del database di destinazione.
2. Scegliete l'applicazione convertita e scegliete Salva.
3. Immettete il percorso della cartella per salvare il codice dell'applicazione convertito e scegliete accedi al database.

Gestione dei progetti di conversione di applicazioni C# in AWS SCT

È possibile aggiungere più progetti di conversione di applicazioni C#, aggiornare il codice dell'applicazione nel AWS SCT progetto o rimuovere un progetto di conversione C# dal AWS SCT progetto.

Per aggiungere un progetto di conversione di applicazioni C# aggiuntivo

1. Espandi il Applicazioni nodo nel pannello di sinistra.
2. Scegli il C# nodo e aprire il menu contestuale (tasto destro del mouse).
3. Scegli New application (Nuova applicazione).
4. Immettere le informazioni necessarie per creare un nuovo progetto di conversione di applicazioni C#. Per ulteriori informazioni, consulta [Creazione di progetti di conversione di applicazioni C#](#).

Dopo aver apportato modifiche al codice dell'applicazione sorgente, caricatelo nel AWS SCT progetto.

Per caricare il codice dell'applicazione aggiornato

1. Espandi il C# nodo sotto Applicazioni nel pannello di sinistra.
2. Scegliere l'applicazione da aggiornare e aprire il menu contestuale (pulsante destro del mouse).
3. Scegliete Aggiorna e poi scegli sì.

AWS SCT carica il codice dell'applicazione dai file sorgente e rimuove i risultati della conversione. Per mantenere le modifiche al codice che hai apportato in AWS SCT e i risultati della conversione, create un nuovo progetto di conversione C#.

Per rimuovere un progetto di conversione di applicazioni C#

1. Espandi il C# nodo sotto Applicazioni nel pannello di sinistra.
2. Scegliere l'applicazione da rimuovere e aprire il menu contestuale (pulsante destro del mouse).
3. Scegliete Eliminare e poi scegli OK.

Creazione di un rapporto di valutazione della conversione delle applicazioni C# in AWS SCT

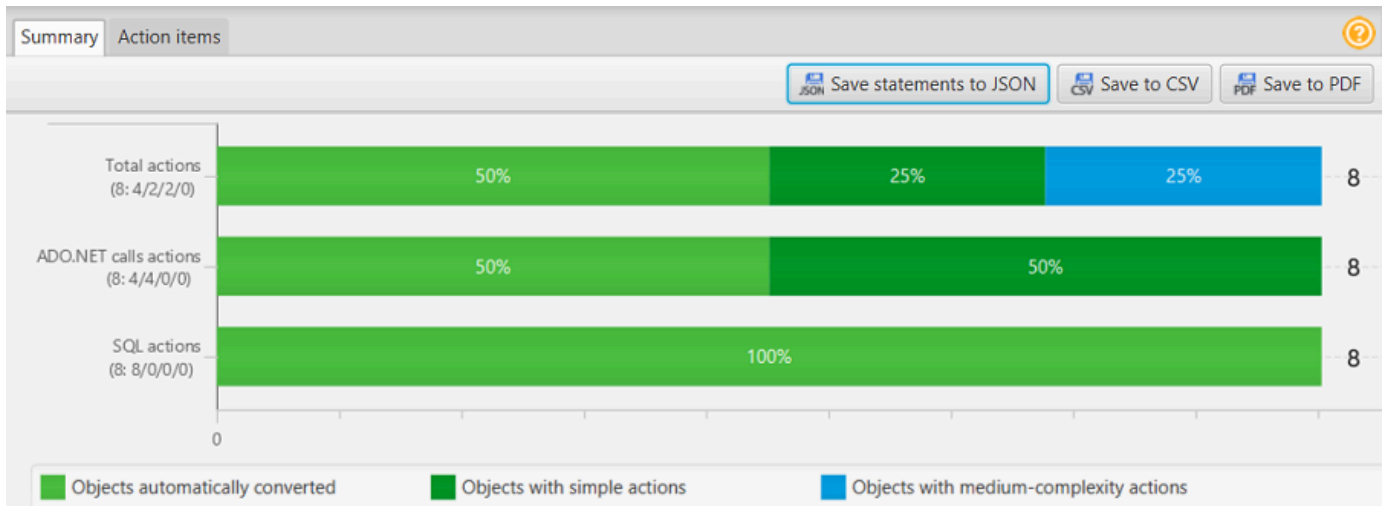
Il Rapporto di valutazione della conversione delle applicazioni C# fornisce informazioni sulla conversione del codice SQL incorporato nell'applicazione C# in un formato compatibile con il database di destinazione. Il rapporto di valutazione fornisce dettagli sulla conversione per tutti i punti di esecuzione SQL e tutti i file di codice sorgente. Il rapporto di valutazione include anche azioni per il codice SQL che AWS SCT impossibile convertire.

Utilizzare la procedura seguente per creare un rapporto di valutazione della conversione delle applicazioni C#.

Per creare un rapporto di valutazione della conversione delle applicazioni C#

1. Espandi il C# nodo sotto Applicazioni nel pannello di sinistra.
2. Scegliere l'applicazione da convertire e aprire il menu contestuale (pulsante destro del mouse).
3. Scegliete Convertire.
4. Sul Visualizza menu, scegli visualizzazione del rapporto di valutazione.
5. Salvataggio del Riepilogo scheda.

Il Riepilogo scheda, mostrata di seguito, mostra le informazioni di riepilogo del rapporto di valutazione dell'applicazione C#. Mostra i risultati della conversione per tutti i punti di esecuzione SQL e tutti i file di codice sorgente.



6. Scegli **Salva le dichiarazioni in JSON** per salvare il codice SQL estratto dall'applicazione C# come file JSON.
7. (Facoltativo) Salvare una copia locale del rapporto come file PDF o come file con valori separati da virgole (CSV):

- Scegli **Salvataggio del PDF** in alto a destra per salvare il rapporto come file PDF.

Il file PDF contiene il riepilogo esecutivo, le azioni da intraprendere e i consigli per la conversione delle applicazioni.

- Scegli **Salvataggio del file** in alto a destra per salvare il rapporto come file CSV.

Il file CSV contiene azioni, azioni consigliate e una stima della complessità dello sforzo manuale necessario per convertire il codice SQL.

Conversione del codice SQL in applicazioni C++ con AWS SCT

Per una conversione da Oracle a PostgreSQL, puoi usare AWS SCT per convertire il codice SQL incorporato nelle applicazioni C++. Questo convertitore di applicazioni C++ specifico comprende la logica dell'applicazione. Raccoglie istruzioni che si trovano in diversi oggetti dell'applicazione, come funzioni, parametri, variabili locali e così via.

Grazie a questa analisi approfondita, il convertitore di codice SQL dell'applicazione C++ fornisce risultati di conversione migliori rispetto al convertitore generico.

Creazione di progetti di conversione di applicazioni C++ inAWS SCT

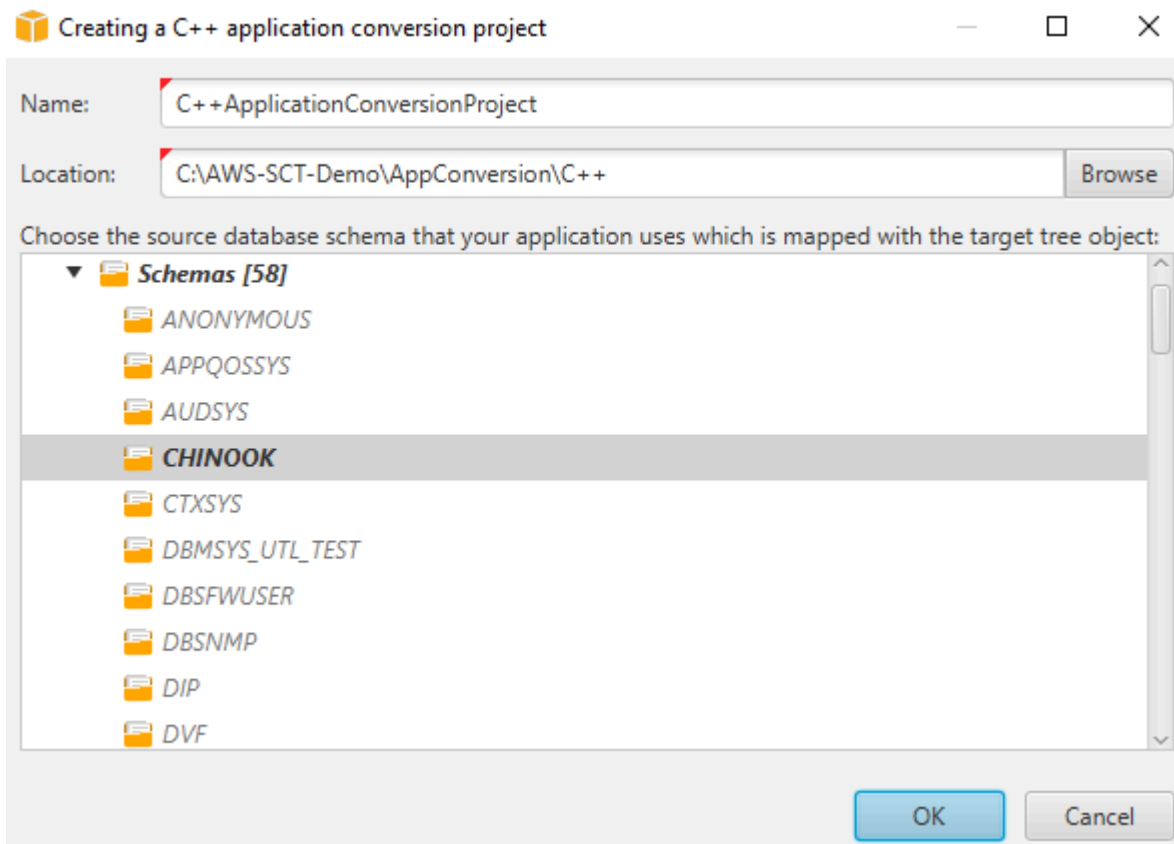
È possibile creare un progetto di conversione di applicazioni C++ solo per convertire gli schemi di database Oracle in schemi di database PostgreSQL. Assicurati di aggiungere una regola di mappatura nel tuo progetto che includa uno schema Oracle di origine e un database PostgreSQL di destinazione. Per ulteriori informazioni, consulta [Creazione di regole di mappatura in AWS SCT](#).

È possibile aggiungere più progetti di conversione di applicazioni in un unicoAWS SCTprogetto.

Per creare un progetto di conversione di applicazioni C++

1. Crea un progetto di conversione del database e aggiungi un database Oracle di origine. Per ulteriori informazioni, consultare [Creare unAWS SCT progetto](#) e [Aggiungere server di database a unAWS SCT progetto](#).
2. Aggiungi una regola di mappatura che includa il tuo database Oracle di origine e un database PostgreSQL di destinazione. È possibile aggiungere un database PostgreSQL di destinazione o utilizzare una piattaforma di database di destinazione PostgreSQL virtuale in una regola di mappatura. Per ulteriori informazioni, consultare [Creazione di regole di mappatura in AWS SCT](#) e [Utilizzo di obiettivi virtuali](#).
3. SulVisualizzamenu, scegliVista principale.
4. SulApplicazionimenu, scegliNuova applicazione C++.

LaCreazione di un progetto di conversione di applicazioni C++viene visualizzata una finestra di dialogo.



5. PerNome, inserire un nome per il progetto di conversione dell'applicazione C++. Poiché ogni progetto di conversione dello schema del database può avere uno o più progetti di conversione di applicazioni secondarie, scegliete un nome che abbia senso se aggiungete più progetti.
6. PerPosizione, inserire la posizione del codice sorgente dell'applicazione.
7. Nell'albero dei sorgenti, scegliete lo schema utilizzato dall'applicazione. Assicuratevi che questo schema faccia parte di una regola di mappatura. AWS SCT evidenzia in grassetto gli schemi che fanno parte di una regola di mappatura.
8. Scegli OK per creare il tuo progetto di conversione di applicazioni C++.
9. Trova il tuo progetto di conversione di applicazioni C++ nel Application nodo nel pannello di sinistra.

Conversione del codice SQL dell'applicazione C++ in AWS SCT

Dopo aver aggiunto l'applicazione C++ al AWS SCT project, converti il codice SQL di questa applicazione in un formato compatibile con la piattaforma di database di destinazione. Utilizzate la seguente procedura per analizzare e convertire il codice SQL incorporato nell'applicazione C++ in AWS SCT.

Per convertire il codice SQL

1. Espandi il C++ nodo sotto Applicazioni nel pannello di sinistra, e scegli l'applicazione da convertire.
2. Nel Progetto di applicazione Oracle di origine, scegli le impostazioni. Rivedi e modifica le impostazioni di conversione per l'applicazione C++ selezionata. È inoltre possibile specificare le impostazioni di conversione per tutte le applicazioni C++ che sono state aggiunte all'AWS SCT progetto. Per ulteriori informazioni, consulta [Gestione dei progetti di conversione di applicazioni C++](#).
3. Per Tipo di compilatore, scegli il compilatore che usi per il codice sorgente dell'applicazione C++ . AWS SCT supporta i seguenti compilatori C++: Microsoft Visual C++, GCC, la collezione GNU Compiler, e Clang. L'opzione di default è Microsoft Visual C++.
4. Per Macro definite dall'utente, inserisci il percorso del file che include le macro definite dall'utente del tuo progetto C++. Accertarsi che il file abbia la seguente struttura: `#define name value`. nell'esempio precedente, `value` è un parametro opzionale. Il valore predefinito per questo parametro opzionale è 1.

Per creare questo file, apri il progetto in Microsoft Visual Studio, quindi scegli Progetto, Proprietà, C/C++, e Preprocessore. Per Definizioni del preprocessore, scegli Modifica e copia nomi e valori in un nuovo file di testo. Quindi, per ogni stringa del file, aggiungi il seguente prefisso: `#define` .

5. Per Elenchi di inclusione esterni, inserisci i percorsi delle cartelle che includono le librerie esterne utilizzate nel tuo progetto C++.
6. Nel riquadro di sinistra (a sinistra), selezionare l'applicazione da convertire e aprire il menu contestuale (pulsante destro del mouse).
7. Scegliete Convertire. AWS SCT analizza i file del codice sorgente, determina la logica dell'applicazione e carica i metadati del codice nel progetto. Questi metadati di codice includono classi C++, oggetti, metodi, variabili globali, interfacce e così via.

nel pannello del database di destinazione, AWS SCT crea una struttura di cartelle simile a quella del progetto dell'applicazione di origine. Qui è possibile esaminare il codice dell'applicazione convertito, come illustrato di seguito.

The screenshot displays two panels of code. The top panel, titled 'Source Oracle file: StringInitialization.cpp', shows C++ code for connecting to an Oracle database and executing a query. The bottom panel, titled 'Target Amazon RDS for PostgreSQL file: StringInitialization.cpp', shows the converted C++ code for connecting to a PostgreSQL database and executing the same query.

```

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          {

```

8. Salvare il codice dell'applicazione convertito. Per ulteriori informazioni, consulta [Salvare il codice dell'applicazione convertito](#).

Salvare il codice dell'applicazione convertito con AWS SCT

Utilizzare la procedura seguente per salvare il codice dell'applicazione convertito.

Per salvare il codice dell'applicazione convertito

1. Espandi il C++ nodo sotto Applicazioni nel pannello del database di destinazione.
2. Scegli la tua applicazione convertita e scegli Salva.
3. Immettete il percorso della cartella per salvare il codice dell'applicazione convertito e scegliete accedi al database.

Gestione di progetti di conversione di applicazioni C++ inAWS SCT

È possibile aggiungere più progetti di conversione di applicazioni C++, modificare le impostazioni di conversione, aggiornare il codice dell'applicazione C++ o rimuovere un progetto di conversione C++ dalAWS SCTprogetto.

Per aggiungere un progetto di conversione di applicazioni C++ aggiuntivo

1. Espandi ilApplicazioninodo nel pannello di sinistra.
2. Scegliete ilC ++nodo e aprire il menu contestuale (tasto destro del mouse).
3. Scegli New application (Nuova applicazione).
4. Immettete le informazioni necessarie per creare un nuovo progetto di conversione di applicazioni C++. Per ulteriori informazioni, consulta [Creazione di progetti di conversione di applicazioni C++](#).

È possibile specificare le impostazioni di conversione per tutti i progetti di conversione di applicazioni C++ presenti nelAWS SCTprogetto.

Per modificare le impostazioni di conversione per tutte le applicazioni C++

1. SullImpostazionimenu, scegliimpostazioni del progetto, e poi scegliConversione di applicazioni.
2. PerTipo di compilatore, scegli il compilatore che usi per il codice sorgente dell'applicazione C++ .AWS SCTsupporta i seguenti compilatori C++:Microsoft Visual C++,GCC, la collezione GNU Compiler, eClang. L'opzione di default èMicrosoft Visual C++.
3. PerMacro definite dall'utente, inserisci il percorso del file che include le macro definite dall'utente del tuo progetto C++. Accertarsi che il file abbia la seguente struttura:`#define name value`. nell'esempio precedente,`value`è un parametro opzionale. Il valore predefinito per questo parametro opzionale è1.

Per creare questo file, apri il progetto in Microsoft Visual Studio, quindi scegliProgetto,Proprietà,C/C++, ePreprocessore. PerDefinizioni del preprocessore, scegliModifica e copia nomi e valori in un nuovo file di testo. Quindi, per ogni stringa del file, aggiungi il seguente prefisso:`#define` .

4. PerElenchi di inclusione esterni, inserisci i percorsi delle cartelle che includono le librerie esterne utilizzate nel tuo progetto C++.
5. SceglieteOKper salvare le impostazioni del progetto e chiudere la finestra.

Oppure è possibile specificare le impostazioni di conversione per ogni progetto di conversione di applicazioni C++. Per ulteriori informazioni, consulta [Conversione del codice SQL dell'applicazione C++](#).

Dopo aver apportato modifiche al codice dell'applicazione sorgente, caricatelo nelAWS SCTprogetto.

Per caricare il codice dell'applicazione aggiornato

1. Espandi ilC ++nodo sottoApplicazioninel pannello di sinistra.
2. Scegliere l'applicazione da aggiornare e aprire il menu contestuale (pulsante destro del mouse).
3. SceglieteAggiornae poi scegli sì.

AWS SCTcarica il codice dell'applicazione dai file sorgente e rimuove i risultati della conversione. Per mantenere le modifiche al codice che hai apportato inAWS SCTe i risultati della conversione, create un nuovo progetto di conversione C++.

InoltreAWS SCTrimuove le impostazioni di conversione dell'applicazione specificate per l'applicazione selezionata. Dopo aver caricato il codice dell'applicazione aggiornato,AWS SCTapplica i valori predefiniti dalle impostazioni del progetto.

Per rimuovere un progetto di conversione di applicazioni C++

1. Espandi ilC ++nodo sottoApplicazioninel pannello di sinistra.
2. Scegliere l'applicazione da rimuovere e aprire il menu contestuale (pulsante destro del mouse).
3. ScegliEliminaree poi scegliOK.

Creazione di un rapporto di valutazione della conversione delle applicazioni C++ inAWS SCT

IlRapporto di valutazione della conversione delle applicazioni C++fornisce informazioni sulla conversione del codice SQL incorporato nell'applicazione C++ in un formato compatibile con il database di destinazione. Il rapporto di valutazione fornisce dettagli sulla conversione per tutti i punti di esecuzione SQL e tutti i file di codice sorgente. Il rapporto di valutazione include anche azioni per il codice SQL cheAWS SCTimpossibile convertire.

Per creare un rapporto di valutazione della conversione delle applicazioni C++

1. Espandi ilC ++nodo sottoApplicazioninel pannello di sinistra.

2. Scegliere l'applicazione da convertire e aprire il menu contestuale (pulsante destro del mouse).
3. Scegliete Convertire.
4. Sul Visualizza menu, scegliete visualizzazione del rapporto di valutazione.
5. Salvataggio del Riepilogo scheda.

La Riepilogo scheda visualizza le informazioni di riepilogo esecutivo tratte dal rapporto di valutazione dell'applicazione C++. Mostra i risultati della conversione per tutti i punti di esecuzione SQL e tutti i file di codice sorgente.

6. Scegli Salva le dichiarazioni in JSON per salvare il codice SQL estratto dall'applicazione Java come file JSON.
7. (Facoltativo) Salvare una copia locale del rapporto come file PDF o come file con valori separati da virgole (CSV):

- Scegliete Salvataggio del PDF in alto a destra per salvare il rapporto come file PDF.

Il file PDF contiene il riepilogo esecutivo, le azioni da intraprendere e i consigli per la conversione delle applicazioni.

- Scegli Salvataggio del file in alto a destra per salvare il rapporto come file CSV.

Il file CSV contiene azioni, azioni consigliate e una stima della complessità dello sforzo manuale necessario per convertire il codice SQL.

Conversione del codice SQL in applicazioni Java con AWS SCT

Per una conversione da Oracle a PostgreSQL, puoi usare AWS Schema Conversion Tool per convertire il codice SQL incorporato nelle applicazioni Java. Questo convertitore di applicazioni Java specifico comprende la logica dell'applicazione. Raccoglie istruzioni che si trovano in diversi oggetti dell'applicazione, come funzioni, parametri, variabili locali e così via.

Grazie a questa analisi approfondita, il convertitore di codice SQL dell'applicazione Java offre risultati di conversione migliori rispetto al convertitore generico.

Se l'applicazione Java utilizza il MyBatis framework per interagire con i database, allora puoi usare AWS SCT per convertire le istruzioni SQL incorporate in MyBatis file e annotazioni XML. Per comprendere la logica di queste istruzioni SQL, AWS SCT utilizza il MyBatis file di configurazione. AWS SCT può scoprire automaticamente questo file nella cartella dell'applicazione oppure è possibile inserire il percorso del file manualmente.

Creazione di progetti di conversione di applicazioni Java in AWS SCT

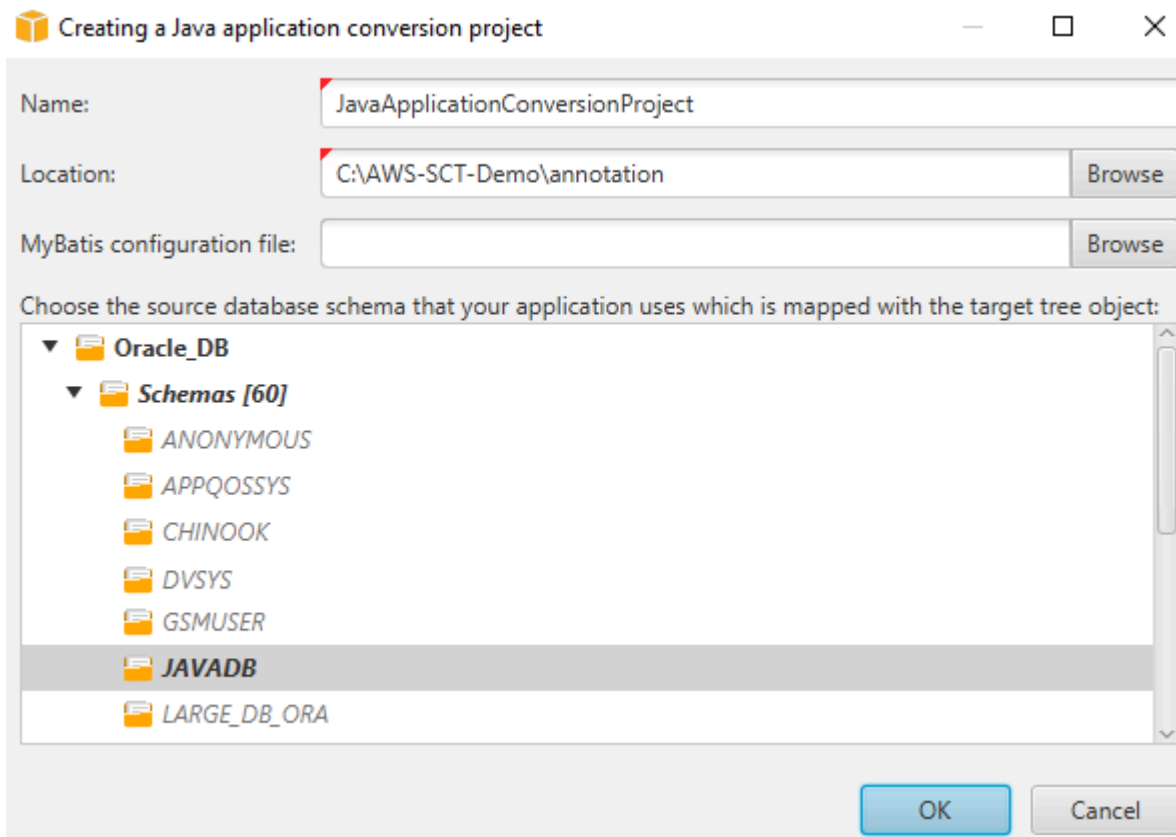
È possibile creare un progetto di conversione di applicazioni Java solo per convertire gli schemi di database Oracle in schemi di database PostgreSQL. Assicurati di aggiungere una regola di mappatura nel tuo progetto che includa uno schema Oracle di origine e un database PostgreSQL di destinazione. Per ulteriori informazioni, consulta [Creazione di regole di mappatura in AWS SCT](#).

È possibile aggiungere più progetti di conversione di applicazioni in un unico AWS SCT progetto. Utilizzare la procedura seguente per creare un progetto di conversione di applicazioni Java (Java).

Per creare un progetto di conversione di applicazioni Java

1. Crea un progetto di conversione del database e aggiungi un database Oracle di origine. Per ulteriori informazioni, consultare [Creare un AWS SCT progetto](#) e [Aggiungere server di database a un AWS SCT progetto](#).
2. Aggiungi una regola di mappatura che includa il tuo database Oracle di origine e un database PostgreSQL di destinazione. È possibile aggiungere un database PostgreSQL di destinazione o utilizzare una piattaforma di database di destinazione PostgreSQL virtuale in una regola di mappatura. Per ulteriori informazioni, consultare [Creazione di regole di mappatura in AWS SCT](#) e [Utilizzo di obiettivi virtuali](#).
3. Sul Visualizza menu, scegli Vista principale.
4. Sul Applicazioni menu, scegli Nuova applicazione Java.

La creazione di un progetto di conversione di applicazioni Java viene visualizzata una finestra di dialogo.



5. PerNome, inserire un nome per il progetto di conversione dell'applicazione Java. Poiché ogni progetto di conversione dello schema del database può avere uno o più progetti di conversione di applicazioni secondarie, scegliete un nome che abbia senso se aggiungete più progetti.
6. PerPosizione, inserire la posizione del codice sorgente dell'applicazione.
7. (Facoltativo) PerMyBatis file di configurazione, immettere il percorso del MyBatis file di configurazione. AWS SCT esegue la scansione della cartella dell'applicazione per scoprire automaticamente questo file. Se questo file non si trova nella cartella dell'applicazione o se utilizzi diversi file di configurazione, inserisci il percorso manualmente.
8. Nell'albero dei sorgenti, scegliete lo schema utilizzato dall'applicazione. Assicurati che questo schema faccia parte di una regola di mappatura. AWS SCT evidenzia in grassetto gli schemi che fanno parte di una regola di mappatura.
9. Scegli OK per creare il tuo progetto di conversione di applicazioni Java.
10. Trova il tuo progetto di conversione di applicazioni Java nell'Applicazioni nodo nel pannello di sinistra.

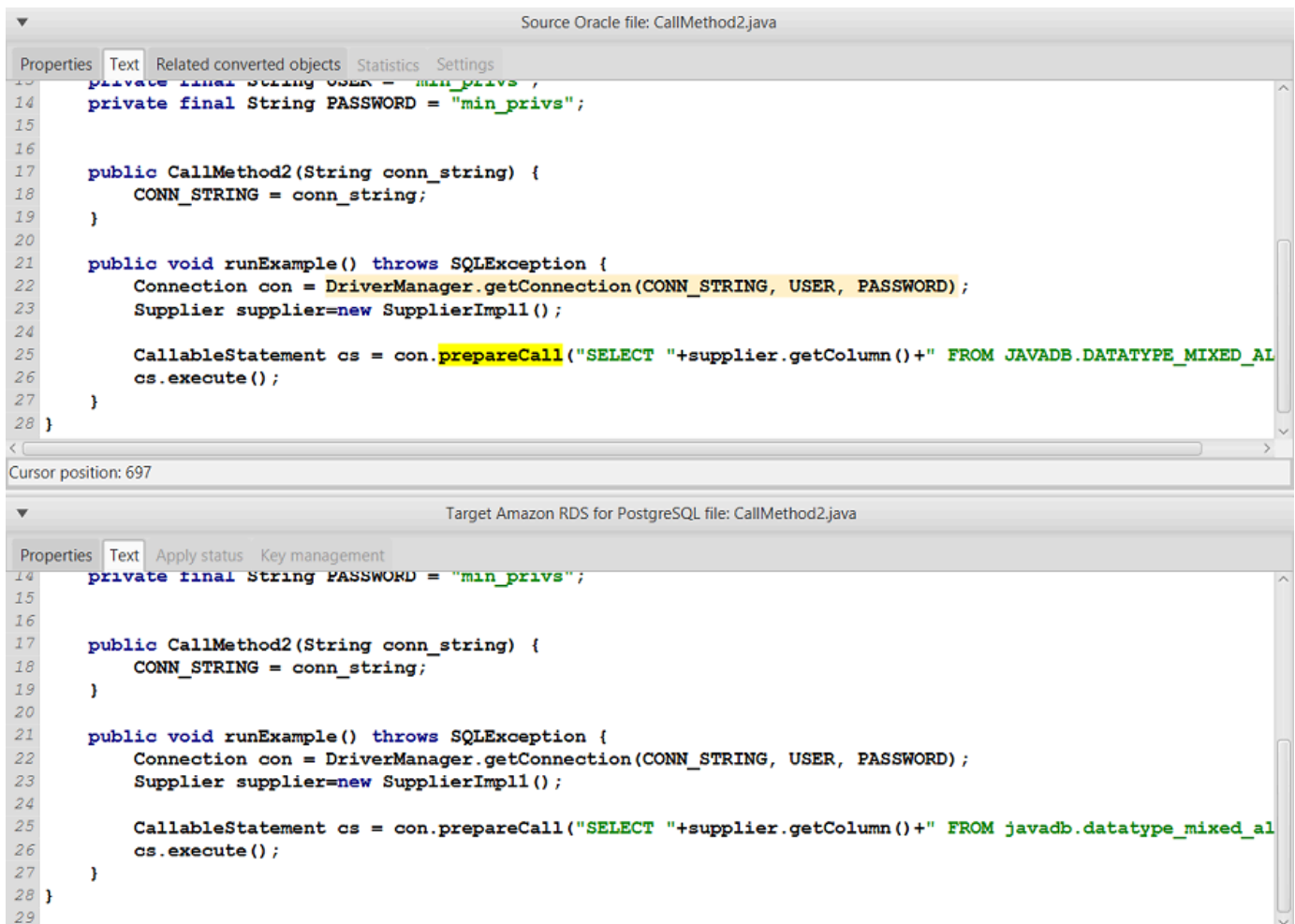
Conversione del codice SQL dell'applicazione Java inAWS SCT

Dopo aver aggiunto l'applicazione Java aAWS SCTproject, converti il codice SQL di questa applicazione in un formato compatibile con la piattaforma di database di destinazione. Utilizzate la seguente procedura per analizzare e convertire il codice SQL incorporato nell'applicazione Java inAWS Schema Conversion Tool.

Per convertire il codice SQL

1. Espandi ilGiavanodo sottoApplicazioninel pannello di sinistra.
2. Scegliere l'applicazione da convertire e aprire il menu contestuale (pulsante destro del mouse).
3. SceglieteConvertire.AWS SCTanalizza i file del codice sorgente, determina la logica dell'applicazione e carica i metadati del codice nel progetto. Questi metadati di codice includono classi Java, oggetti, metodi, variabili globali, interfacce e così via.

nel pannello del database di destinazione,AWS SCTcrea una struttura di cartelle simile a quella del progetto dell'applicazione di origine. Qui puoi rivedere il codice dell'applicazione convertito.



The screenshot displays two windows from the AWS Schema Conversion Tool. The top window, titled 'Source Oracle file: CallMethod2.java', shows the original Java code. The bottom window, titled 'Target Amazon RDS for PostgreSQL file: CallMethod2.java', shows the converted code. The conversion process has modified the SQL query in the `runExample()` method to be compatible with PostgreSQL.

```
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 }
```

4. Salvare il codice dell'applicazione convertito. Per ulteriori informazioni, consulta [Salvare il codice dell'applicazione convertito](#).

Le applicazioni Java potrebbero includere codice SQL che interagisce con diversi database di origine. È possibile migrare a PostgreSQL diversi di questi database di origine. In questo caso, assicurati di non convertire il codice SQL che interagisce con i database che hai escluso dall'ambito della migrazione. È possibile escludere i file sorgente dell'applicazione Java dall'ambito di conversione. A tale scopo, deselezionate le caselle di controllo relative ai nomi dei file che desiderate escludere dall'ambito di conversione.

Dopo aver modificato l'ambito di conversione, AWS SCT analizza comunque il codice SQL di tutti i file sorgente delle applicazioni Java. Quindi, AWS SCT copia nella cartella di destinazione tutti i file di origine esclusi dall'ambito di conversione. Questa operazione consente di creare l'applicazione dopo aver salvato i file dell'applicazione convertiti.

Salvare il codice dell'applicazione convertito con AWS SCT

Utilizzare la procedura seguente per salvare il codice dell'applicazione convertito.

Per salvare il codice dell'applicazione convertito

1. Espandi il Giavano sotto Applicazioni nel pannello del database di destinazione.
2. Scegliete l'applicazione convertita e scegliete Salva.
3. Immettete il percorso della cartella per salvare il codice dell'applicazione convertito e scegliete accedi al database.

Se l'applicazione Java di origine utilizza MyBatis framework, assicuratevi di aggiornare il file di configurazione per utilizzarlo con il nuovo database.

Gestione dei progetti di conversione di applicazioni Java in AWS SCT

È possibile aggiungere più progetti di conversione di applicazioni Java, aggiornare il codice dell'applicazione nel AWS SCT progetto o rimuovere un progetto di conversione Java dal AWS SCT progetto.

Per aggiungere un altro progetto di conversione di applicazioni Java

1. Espandi il Applicazioni nodo nel pannello di sinistra.
2. Scegli il Giavano e aprire il menu contestuale (tasto destro del mouse).
3. Scegli New application (Nuova applicazione).
4. Immettete le informazioni necessarie per creare un nuovo progetto di conversione di applicazioni Java. Per ulteriori informazioni, consulta [Creazione di progetti di conversione di applicazioni Java](#).

Dopo aver apportato modifiche al codice dell'applicazione sorgente, caricatelo nel AWS SCT progetto.

Per caricare il codice dell'applicazione aggiornato

1. Espandi il Giavano sotto Applicazioni nel pannello di sinistra.
2. Scegliere l'applicazione da aggiornare e aprire il menu contestuale (pulsante destro del mouse).
3. Scegliete Aggiorna e poi scegli sì.

AWS SCT carica il codice dell'applicazione dai file sorgente e rimuove i risultati della conversione. Per mantenere le modifiche al codice che hai apportato in AWS SCT e i risultati della conversione, crea un nuovo progetto di conversione Java.

Se l'applicazione Java di origine utilizza il MyBatis struttura AWS SCT utilizza il MyBatis file di configurazione per analizzare il codice SQL. Dopo aver modificato questo file, caricalo nel AWS SCT progetto.

Per modificare il percorso di MyBatis file di configurazione

1. Espandi il Giavano sotto Applicazioni nel pannello di sinistra.
2. Scegli la tua applicazione, quindi scegli le impostazioni.
3. Scegli Sfoglia, quindi scegli MyBatis file di configurazione.
4. Seleziona Apply (Applica).
5. Nel pannello di sinistra (a sinistra), scegliere l'applicazione e aprire il menu contestuale (pulsante destro del mouse), quindi selezionare Aggiorna.

Per rimuovere un progetto di conversione di applicazioni Java

1. Espandi il Giavano sotto Applicazioni nel pannello di sinistra.
2. Scegliere l'applicazione da rimuovere e aprire il menu contestuale (pulsante destro del mouse).
3. Scegli Elimina e poi scegli OK.

Creazione di un rapporto di valutazione della conversione delle applicazioni Java in AWS SCT

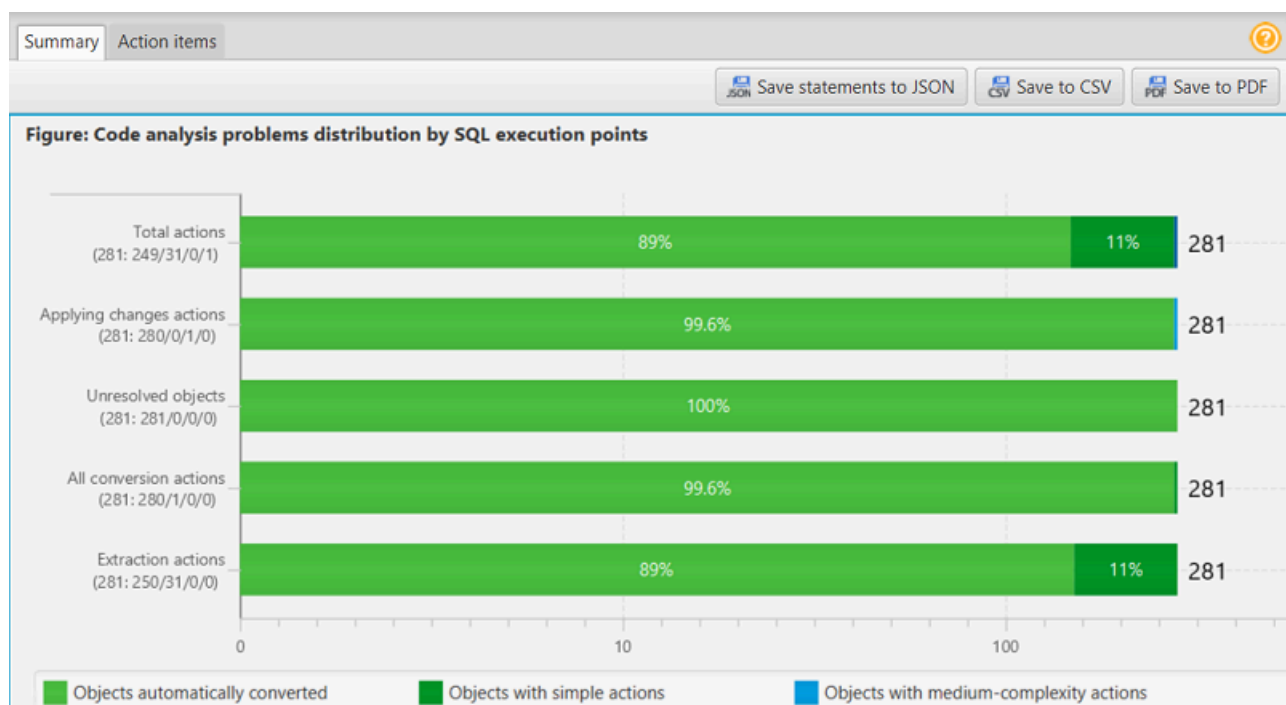
Il Rapporto di valutazione della conversione delle applicazioni Java fornisce informazioni sulla conversione del codice SQL incorporato nell'applicazione Java in un formato compatibile con il database di destinazione. Il rapporto di valutazione fornisce dettagli sulla conversione per tutti i punti di esecuzione SQL e tutti i file di codice sorgente. Il rapporto di valutazione include anche azioni per il codice SQL che AWS SCT impossibile convertire.

Utilizzare la procedura seguente per creare un rapporto di valutazione della conversione delle applicazioni Java (Computer).

Per creare un rapporto di valutazione della conversione delle applicazioni Java

1. Espandi ilGiavanodo sottoApplicazioninel pannello di sinistra.
2. Scegliere l'applicazione da convertire e aprire il menu contestuale (pulsante destro del mouse).
3. ScegliConvertire.
4. SulVisualizzamenu, sceglivisualizzazione del rapporto di valutazione.
5. Verifica la scheda Summary (Riepilogo).

LaRiepilogola scheda, mostrata di seguito, mostra le informazioni di riepilogo esecutivo tratte dal rapporto di valutazione dell'applicazione Java. Mostra i risultati della conversione per tutti i punti di esecuzione SQL e tutti i file di codice sorgente.



6. ScegliSalva le dichiarazioni in JSONper salvare il codice SQL estratto dall'applicazione Java come file JSON.
7. (Facoltativo) Salvare una copia locale del rapporto come file PDF o come file con valori separati da virgole (CSV):

- SceglieteSalvataggio del PDFin alto a destra per salvare il rapporto come file PDF.

Il file PDF contiene il riepilogo esecutivo, le azioni da intraprendere e i consigli per la conversione delle applicazioni.

- SceglieteSalvataggio del filein alto a destra per salvare il rapporto come file CSV.

Il file CSV contiene azioni, azioni consigliate e una stima della complessità dello sforzo manuale necessario per convertire il codice SQL.

Conversione del codice SQL nelle applicazioni Pro*C conAWS SCT

Per una conversione da Oracle a PostgreSQL, puoi usareAWS Schema Conversion Tool(AWS SCT) per convertire il codice SQL incorporato nelle applicazioni Pro*C. Questo convertitore di applicazioni Pro*C specifico comprende la logica dell'applicazione. Raccoglie istruzioni che si trovano in diversi oggetti dell'applicazione, come funzioni, parametri, variabili locali e così via.

Grazie a questa analisi approfondita, il convertitore di codice SQL dell'applicazione Pro*C offre risultati di conversione migliori rispetto al convertitore generico.

Creazione di progetti di conversione di applicazioni Pro*C inAWS SCT

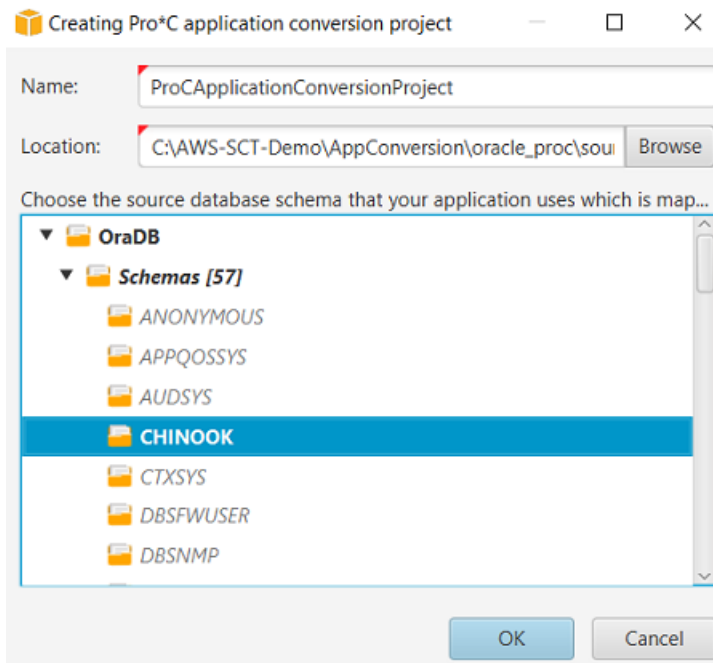
È possibile creare un progetto di conversione di applicazioni Pro*C solo per convertire gli schemi di database Oracle in schemi di database PostgreSQL. Assicurati di aggiungere una regola di mappatura nel tuo progetto che includa uno schema Oracle di origine e un database PostgreSQL di destinazione. Per ulteriori informazioni, consulta [Creazione di regole di mappatura in AWS SCT](#).

È possibile aggiungere più progetti di conversione di applicazioni in un unicoAWS SCTprogetto. Utilizzare la procedura seguente per creare un progetto di conversione dell'applicazione Pro*C (Pro*C).

Per creare un progetto di conversione di applicazioni Pro*C

1. Creare un progetto di conversione del database e aggiungere un database Oracle di origine. Per ulteriori informazioni, consultare [Creare unAWS SCT progetto](#) e [Aggiungere server di database a unAWS SCT progetto](#).
2. Aggiungi una regola di mappatura che includa il tuo database Oracle di origine e un database PostgreSQL di destinazione. È possibile aggiungere un database PostgreSQL di destinazione o utilizzare una piattaforma di database di destinazione PostgreSQL virtuale in una regola di mappatura. Per ulteriori informazioni, consultare [Creazione di regole di mappatura in AWS SCT](#) e [Utilizzo di obiettivi virtuali](#).
3. SulVisualizzamenu, scegliVista principale.
4. SulApplicazionimenu, scegliNuova applicazione Pro*C.

La Creazione di un progetto di conversione di applicazioni Pro*C viene visualizzata una finestra di dialogo.



5. Per Nome, inserire un nome per il progetto di conversione dell'applicazione Pro*C. Poiché ogni progetto di conversione dello schema del database può avere uno o più progetti di conversione di applicazioni secondarie, scegliete un nome che abbia senso se aggiungete più progetti.
6. Per Posizione, inserire la posizione del codice sorgente dell'applicazione.
7. Nell'albero dei sorgenti, scegliete lo schema utilizzato dall'applicazione. Assicuratevi che questo schema faccia parte di una regola di mappatura. AWS SCT evidenzia in grassetto gli schemi che fanno parte di una regola di mappatura.
8. Scegli OK per creare il progetto di conversione dell'applicazione Pro*C.
9. Trovate il vostro progetto di conversione delle applicazioni Pro*C nel Application nodo nel pannello di sinistra.

Conversione del codice SQL dell'applicazione Pro*C in AWS SCT

Dopo aver aggiunto l'applicazione Pro*C al AWS SCT project, converti il codice SQL di questa applicazione in un formato compatibile con la piattaforma di database di destinazione. Utilizzate la seguente procedura per analizzare e convertire il codice SQL incorporato nell'applicazione Pro*C nel AWS Schema Conversion Tool.

Per convertire il codice SQL

1. Espandi ilPro*Cnodo sottoApplicazioneinil pannello di sinistra.
2. Scegli l'applicazione da convertire e poi scegliImpostazioni.
 - a. Perpercorso del file di intestazione globale, inserite il percorso dei file di intestazione utilizzati dal progetto dell'applicazione.
 - b. SceglieteInterpreta tutte le variabili host non risolte comeper visualizzare tutte le variabili non risolte nel codice convertito.
 - c. ScegliUsa la funzione di conversione delle stringhe a larghezza fissa del pacchetto di estensioneper utilizzare le funzioni del pacchetto di estensione nel codice SQL convertito.AWS SCTinclude i file del pacchetto di estensione nel progetto dell'applicazione.
 - d. ScegliTrasforma blocchi PL/SQL anonimi in chiamate SQL autonome o funzioni memorizzateper creare procedure memorizzate nel database di destinazione per tutti i blocchi PL/SQL anonimi.AWS SCTinclude quindi le esecuzioni di queste stored procedure nel codice dell'applicazione convertito.
 - e. ScegliUsa il flusso del cursore personalizzatoper migliorare la conversione dei cursori del database Oracle.
3. Nel pannello di sinistra (a sinistra), scegli l'applicazione da convertire e apri il menu contestuale (fai clic con il pulsante destro del mouse).
4. SceglieteConvertire.AWS SCTanalizza i file del codice sorgente, determina la logica dell'applicazione e carica i metadati del codice nel progetto. Questi metadati di codice includono classi Pro*C, oggetti, metodi, variabili globali, interfacce e così via.

nel pannello del database di destinazione,AWS SCTcrea una struttura di cartelle simile a quella del progetto dell'applicazione di origine. Qui puoi rivedere il codice dell'applicazione convertito.

The screenshot displays two panels of code. The top panel, titled 'Source Oracle app function: main() int', shows the original Oracle Pro*C code with line numbers 11 through 20. The code includes a variable declaration, a comment, and SQL execution statements. The bottom panel, titled 'Target Amazon RDS for PostgreSQL app function: main() int', shows the converted PostgreSQL code with line numbers 8 through 18. The converted code uses 'EXEC SQL' statements and standard SQL syntax.

```

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. Salvare il codice dell'applicazione convertito. Per ulteriori informazioni, consulta [Modifica e salvataggio del codice dell'applicazione convertito](#).

Modifica e salvataggio del codice dell'applicazione convertito con AWS SCT

È possibile modificare le istruzioni SQL convertite e utilizzare AWS SCT per incorporare questo codice modificato nel codice dell'applicazione Pro*C convertito. Utilizzare la procedura seguente per modificare il codice SQL convertito.

Per modificare il codice SQL convertito

1. Espandi il Pro*C nodo sotto Applicazioni nel pannello di sinistra.
2. Scegliere l'applicazione da convertire (applicazione), aprire il menu contestuale (pulsante destro del mouse), quindi selezionare Convertire.
3. Sul Visualizza menu, scegliere visualizzazione del rapporto di valutazione.
4. Scegli accedi al file CSV per salvare il codice SQL estratto dall'applicazione Pro*C come file CSV.
5. Immettete il nome del file CSV per salvare il codice SQL estratto e scegliete Salva.
6. Modifica il codice SQL estratto.
7. Sul Visualizza menu, scegli Vista principale.
8. Espandi il Pro*C nodo sotto Applicazioni nel pannello del database di destinazione.

9. Scegliere l'applicazione convertita (l'applicazione), aprire il menu contestuale (pulsante destro del mouse), quindi selezionare **Importa dichiarazioni da CSV**.
10. Scegliere, quindi scegliere il file con il codice SQL modificato e scegliere **Apri**.

AWS SCT suddivide le istruzioni SQL convertite in parti e le colloca negli oggetti appropriati del codice dell'applicazione sorgente. Utilizzare la procedura seguente per salvare il codice dell'applicazione convertito.

Per salvare il codice dell'applicazione convertito

1. Espandi il **Pro*C** nodo sotto **Applicazioni** nel pannello del database di destinazione.
2. Scegliete l'applicazione convertita e scegliete **Salva**.
3. Immettete il percorso della cartella per salvare il codice dell'applicazione convertito e scegliete **Accedi al database**.

Gestione dei progetti di conversione di applicazioni Pro*C in AWS SCT

È possibile aggiungere più progetti di conversione di applicazioni Pro*C, aggiornare il codice dell'applicazione nel AWS SCT progetto o rimuovere un progetto di conversione Pro*C dal AWS SCT progetto.

Per aggiungere un ulteriore progetto di conversione dell'applicazione Pro*C

1. Espandi il **Applicazioni** nodo nel pannello di sinistra.
2. Scegli il **Pro*C** nodo e aprire il menu contestuale (tasto destro del mouse).
3. Scegli **New application** (Nuova applicazione).
4. Immettete le informazioni necessarie per creare un nuovo progetto di conversione dell'applicazione Pro*C. Per ulteriori informazioni, consulta [Creazione di progetti di conversione di applicazioni Pro*C](#).

Dopo aver apportato modifiche al codice dell'applicazione sorgente, caricatelo nel AWS SCT progetto.

Per caricare il codice dell'applicazione aggiornato

1. Espandi il **Pro*C** nodo sotto **Applicazioni** nel pannello di sinistra.
2. Scegliere l'applicazione da aggiornare e aprire il menu contestuale (pulsante destro del mouse).

3. Scegliete Aggiorna e poi scegli Sì.

AWS SCT carica il codice dell'applicazione dai file sorgente e rimuove i risultati della conversione. Per mantenere le modifiche al codice che hai apportato in AWS SCT e i risultati della conversione, create un nuovo progetto di conversione Pro*C.

Per rimuovere un progetto di conversione di applicazioni Pro*C

1. Espandi il Pro*C nodo sotto Applicazioni nel pannello di sinistra.
2. Scegliere l'applicazione da rimuovere e aprire il menu contestuale (pulsante destro del mouse).
3. Scegliete Elimina e poi scegli OK.

Creazione di un rapporto di valutazione della conversione delle applicazioni Pro*C in AWS SCT

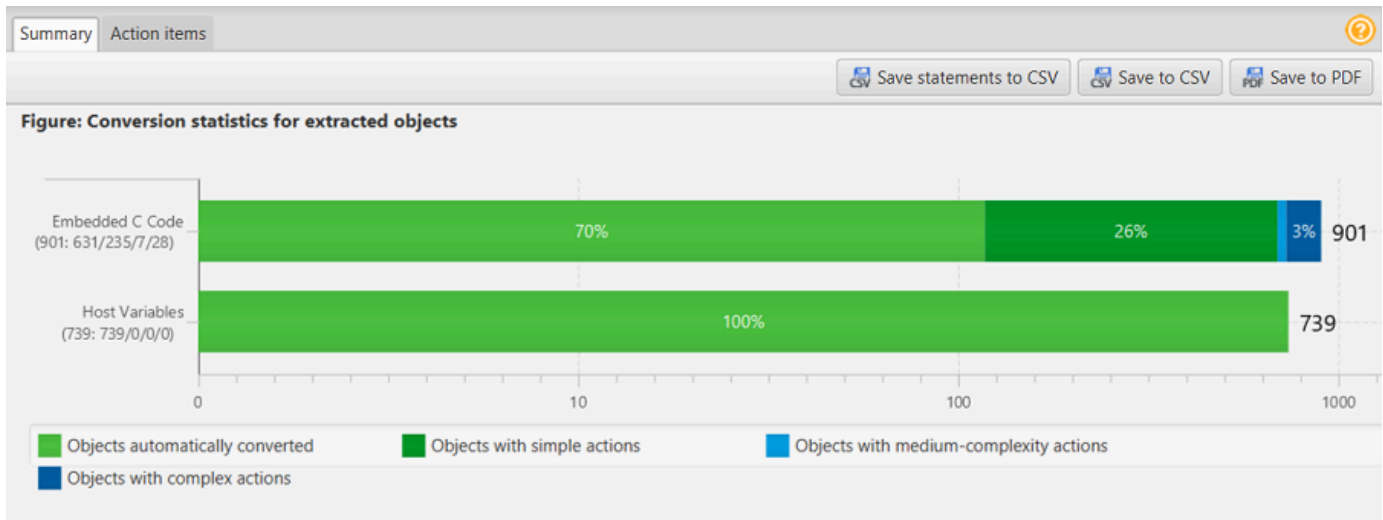
Il Rapporto di valutazione della conversione delle applicazioni Pro*C fornisce informazioni sulla conversione del codice SQL incorporato nell'applicazione Pro*C in un formato compatibile con il database di destinazione. Il rapporto di valutazione fornisce dettagli sulla conversione per tutti i punti di esecuzione SQL e tutti i file di codice sorgente. Il rapporto di valutazione include anche azioni per il codice SQL che AWS SCT impossibile convertire.

Utilizzare la procedura seguente per creare un rapporto di valutazione della conversione delle applicazioni Pro*C (Pro*C).

Per creare un rapporto di valutazione della conversione delle applicazioni Pro*C

1. Espandi il Pro*C nodo sotto Applicazioni nel pannello di sinistra.
2. Scegliere l'applicazione da convertire e aprire il menu contestuale (pulsante destro del mouse).
3. Scegli Convertire.
4. Sul Visualizza menu, scegli visualizzazione del rapporto di valutazione.
5. Verifica la scheda Summary (Riepilogo).

La Riepilogo scheda, mostrata di seguito, mostra le informazioni riassuntive tratte dal rapporto di valutazione delle applicazioni Pro*C. Mostra i risultati della conversione per tutti i punti di esecuzione SQL e tutti i file di codice sorgente.



6. Scegliaci il file CSV per salvare il codice SQL estratto dall'applicazione Pro*C come file con valori separati da virgole (CSV).
7. (Facoltativo) Salvare una copia locale del rapporto come file PDF o come file con valori separati da virgole (CSV):

- Scegli Salvataggio del PDF in alto a destra per salvare il rapporto come file PDF.

Il file PDF contiene il riepilogo esecutivo, le azioni da intraprendere e i consigli per la conversione delle applicazioni.

- Scegliete Salvataggio del file in alto a destra per salvare il rapporto come file CSV.

Il file CSV contiene azioni, azioni consigliate e una stima della complessità dello sforzo manuale necessario per convertire il codice SQL.

Utilizzo dei pacchetti di AWS SCT estensione

Un pacchetto di AWS SCT estensione è un modulo aggiuntivo che emula le funzioni presenti in un database di origine necessarie per la conversione degli oggetti nel database di destinazione. Prima di poter installare un pacchetto di AWS SCT estensione, è necessario convertire lo schema del database.

Ogni pacchetto di AWS SCT estensione include i seguenti componenti:

- Schema DB: include funzioni, procedure e tabelle SQL per l'emulazione di determinati oggetti di database OLTP (Online Transaction Processing) e OLAP (Online Analytical Processing), come le sequenze. Inoltre, emula non built-in-functions è supportata dal database di origine. Il nome di questo schema ha il seguente formato: `aws_`*database_engine_name*`_ext`
- AWS Lambda funzioni (per alcuni database OLTP): include AWS Lambda funzioni che emulano funzionalità di database complesse, come la pianificazione dei processi e l'invio di e-mail.
- Librerie personalizzate per database OLAP: include un set di librerie Java e Python che è possibile utilizzare per migrare gli script di estrazione, trasformazione e caricamento (ETL) di Microsoft SQL Server Integration Services (SSIS) verso o. AWS Glue AWS Glue Studio

Le librerie Java includono i seguenti moduli:

- `spark-excel_2.11-0.13.1.jar`— Per emulare la funzionalità dei componenti di origine e destinazione di Excel.
- `spark-xml_2.11-0.9.0.jar`, `poi-ooxml-schemas-4.1.2.jar`, e `xmlbeans-3.1.0.jar` — Per emulare la funzionalità del componente di origine XML.

Le librerie Python includono i seguenti moduli:

- `sct_utils.py`— Per emulare i tipi di dati di origine e preparare i parametri per la query SQL Spark.
- `ssis_datetime.py`— Per emulare le funzioni integrate di data e ora.
- `ssis_null.py`— Per emulare le ISNULL REPLACENULL funzioni integrate.
- `ssis_string.py`— Per emulare le funzioni integrate nelle stringhe.

Per ulteriori informazioni su queste librerie, consulta [Utilizzo di librerie personalizzate per i pacchetti di AWS SCT estensione](#).

È possibile applicare i pacchetti di AWS SCT estensione in due modi:

- AWS SCT può applicare automaticamente un pacchetto di estensione quando si applica uno script di database di destinazione scegliendo Applica al database dal menu contestuale. AWS SCT applica il pacchetto di estensione prima di applicare tutti gli altri oggetti dello schema.
- Per applicare manualmente un pacchetto di estensione, scegli il database di destinazione, quindi scegli Applica pacchetto di estensione per dal menu contestuale (fai clic con il pulsante destro del mouse). Nella maggior parte dei casi, l'applicazione automatica è sufficiente. Tuttavia, potresti voler applicare il pacchetto manualmente se viene eliminato accidentalmente.

Ogni volta che si applica un pacchetto di AWS SCT estensione a un data store di destinazione, i componenti vengono AWS SCT sovrascritti e viene visualizzata una notifica al riguardo. Per disattivare queste notifiche, scegli Impostazioni, Impostazioni globali, Notifiche, quindi seleziona Nascondi l'avviso di sostituzione del pacchetto di estensione.

Per una conversione da Microsoft SQL Server a PostgreSQL, puoi utilizzare il pacchetto di estensione da SQL Server a PostgreSQL. AWS SCT Questo pacchetto di estensione emula SQL Server Agent e SQL Server Database Mail. Per ulteriori informazioni, consultare [Emulazione di SQL Server Agent in PostgreSQL con un pacchetto di estensione](#) e [Emulazione di SQL Server Database Mail in PostgreSQL con un pacchetto di estensione](#).

Di seguito sono disponibili ulteriori informazioni sull'utilizzo dei pacchetti di AWS SCT estensione.

Argomenti

- [Autorizzazioni per l'utilizzo del pacchetto di AWS SCT estensioni](#)
- [Utilizzo dello schema del pacchetto di estensione](#)
- [Utilizzo di librerie personalizzate per i pacchetti di AWS SCT estensione](#)
- [Utilizzo delle AWS Lambda funzioni del pacchetto di AWS SCT estensione](#)
- [Configurazione delle funzioni per il pacchetto di estensione AWS SCT](#)

Autorizzazioni per l'utilizzo del pacchetto di AWS SCT estensioni

Il pacchetto di AWS SCT estensione per Amazon Aurora emula l'invio di posta, la pianificazione dei lavori, l'accodamento e altre operazioni utilizzando le funzioni. AWS Lambda Quando applichi il pacchetto di AWS SCT estensione al database Aurora di destinazione, AWS SCT crea un nuovo ruolo AWS Identity and Access Management (IAM) e una policy IAM in linea. Successivamente, AWS SCT crea una nuova funzione Lambda e configura il cluster Aurora DB per le connessioni in uscita

a. AWS Lambda Per eseguire queste operazioni, assicurati di concedere le seguenti autorizzazioni richieste al tuo utente IAM:

- `iam:CreateRole`— per creare un nuovo ruolo IAM per il tuo AWS account.
- `iam:CreatePolicy`— per creare una nuova policy IAM per il tuo AWS account.
- `iam:AttachRolePolicy`— per allegare la policy specificata al tuo ruolo IAM.
- `iam:PutRolePolicy`— per aggiornare un documento di policy in linea incorporato nel tuo ruolo IAM.
- `iam:PassRole`— per passare il ruolo IAM specificato al motore delle regole.
- `iam:TagRole`— per aggiungere tag a un ruolo IAM.
- `iam:TagPolicy`— per aggiungere tag a una policy IAM.
- `lambda:ListFunctions`— per visualizzare l'elenco delle tue funzioni Lambda.
- `lambda:ListTags`— per visualizzare l'elenco dei tag delle funzioni Lambda.
- `lambda:CreateFunction`— per creare una nuova funzione Lambda.
- `rds:AddRoleToDBCluster`— per associare un ruolo IAM al cluster Aurora DB.

Il pacchetto di AWS SCT estensione per Amazon Redshift emula le funzioni di base del data warehouse di origine necessarie per applicare oggetti convertiti ad Amazon Redshift. Prima di applicare il codice convertito ad Amazon Redshift, devi applicare il pacchetto di estensione per Amazon Redshift. A tal fine, includi `iam:SimulatePrincipalPolicy` nella tua policy IAM.

AWS SCT utilizza IAM Policy Simulator per verificare le autorizzazioni richieste per l'installazione del pacchetto di estensione Amazon Redshift. IAM Policy Simulator può visualizzare un messaggio di errore anche se hai configurato correttamente il tuo utente IAM. Si tratta di un problema noto di IAM Policy Simulator. Inoltre, IAM Policy Simulator visualizza un messaggio di errore quando `iam:SimulatePrincipalPolicy` non è inclusa nella policy IAM. In questi casi, puoi ignorare il messaggio di errore e applicare il pacchetto di estensione utilizzando la procedura guidata del pacchetto di estensione. Per ulteriori informazioni, consulta [Applicazione del pacchetto di estensione](#).

Utilizzo dello schema del pacchetto di estensione

Quando converti lo schema del database o del data warehouse, AWS SCT aggiunge un ulteriore schema per il database di destinazione. Questo schema implementa le funzioni di sistema SQL del

database di origine richieste durante la scrittura dello schema convertito nel database di destinazione. Lo schema aggiuntivo viene chiamato schema del pacchetto di estensione.

Lo schema del pacchetto di estensione per i database OLTP viene denominato in base al database di origine come segue:

- Microsoft SQL Server: `AWS_SQLSERVER_EXT`
- MySQL: `AWS_MYSQL_EXT`
- Oracle: `AWS_ORACLE_EXT`
- PostgreSQL: `AWS_POSTGRESQL_EXT`

Lo schema del pacchetto di estensione per le applicazioni data warehouse OLAP viene denominato in base al data store di origine come segue:

- Greenplum: `AWS_GREENPLUM_EXT`
- Microsoft SQL Server: `AWS_SQLSERVER_EXT`
- Netezza: `AWS_NETEZZA_EXT`
- Oracle: `AWS_ORACLE_EXT`
- Teradata: `AWS_TERADATA_EXT`
- Vertica: `AWS_VERTICA_EXT`

Utilizzo di librerie personalizzate per i pacchetti di AWS SCT estensione

In alcuni casi, non è AWS SCT possibile convertire le funzionalità del database di origine in funzionalità equivalenti nel database di destinazione. Il pacchetto di AWS SCT estensione pertinente contiene librerie personalizzate che emulano alcune funzionalità del database di origine nel database di destinazione.

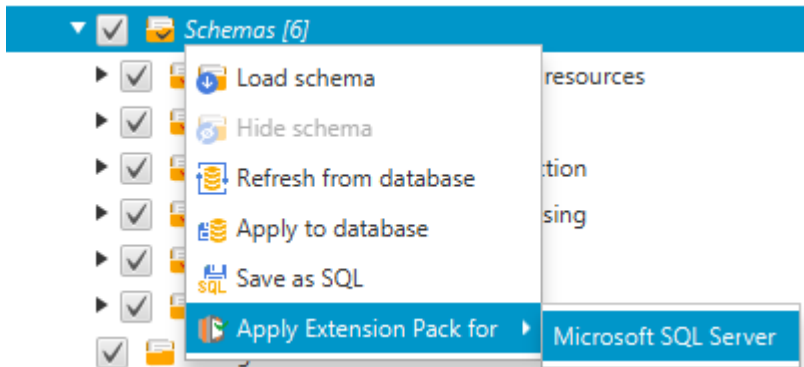
Se stai convertendo un database transazionale, vedi. [Utilizzo delle AWS Lambda funzioni del pacchetto di AWS SCT estensione](#)

Applicazione del pacchetto di estensione

È possibile applicare il pacchetto di AWS SCT estensione utilizzando la procedura guidata del pacchetto di estensione o quando si applica il codice convertito al database di destinazione.

Per applicare il pacchetto di estensione utilizzando la procedura guidata per il pacchetto di estensione

1. Nell'albero del database di destinazione AWS Schema Conversion Tool, apri il menu contestuale (fai clic con il pulsante destro del mouse), scegli Applica il pacchetto di estensione per, quindi scegli la piattaforma di database di origine.



Viene visualizzata la procedura guidata del pacchetto di estensione.

2. Leggi la pagina Welcome (Benvenuti) e scegli Next (Avanti).
3. Nella pagina AWS delle impostazioni del profilo, procedi come segue:
 - Se stai reinstallando solo lo schema del pacchetto di estensione, scegli Skip this step for now (Salta questa fase per ora) e seleziona Next (Avanti). L'opzione Ignora questo passaggio per ora è disponibile solo per i database OLTP (Online Transaction Processing).
 - Se stai caricando una nuova libreria, fornisci le credenziali per connetterti alla tua. Account AWS Utilizzate questo passaggio solo quando convertite database OLAP o script ETL. Puoi usare le tue credenziali AWS Command Line Interface (AWS CLI) se le hai installate. AWS CLI Puoi anche utilizzare le credenziali che hai precedentemente archiviato in un profilo nelle impostazioni globali dell'applicazione e associate al progetto. Se necessario, scegliete Passa alle impostazioni globali per configurare o associare un profilo diverso al AWS SCT progetto. Per ulteriori informazioni, consulta [Archiviazione dei profili AWS di servizio in AWS SCT](#).
4. Se stai caricando una nuova libreria, scegli Devo caricare una libreria nella pagina di caricamento della libreria. Utilizzate questo passaggio solo quando convertite database OLAP o script ETL. Quindi, fornisci il percorso Amazon S3, quindi scegli Carica libreria su S3.

Se hai già caricato la libreria, scegli Ho già caricato le librerie, usa il mio bucket S3 esistente nella pagina di caricamento della libreria. Quindi, fornisci il percorso Amazon S3.

Al termine, seleziona Next (Avanti).

5. Nella pagina di emulazione delle funzioni, scegli Crea pacchetto di estensione. I messaggi vengono visualizzati con lo stato delle operazioni del pacchetto di estensione.

Al termine, seleziona Finish (Fine).

Per applicare il pacchetto di estensione quando si applica il codice convertito

1. Specificate il bucket Amazon S3 nel vostro AWS profilo di servizio. Utilizza questo passaggio solo quando converti database OLAP o script ETL. Per ulteriori informazioni, consulta [Archiviazione dei profili AWS di servizio in AWS SCT](#).

Assicurati che la tua policy sui bucket di Amazon S3 includa le seguenti autorizzazioni:

```
{
  "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"]
    }
  ]
}
```

Nell'esempio precedente, sostituisci *DataExtractionAgentName111122223333:user/* con il nome del tuo utente IAM.

2. Converti gli schemi del data warehouse di origine. Per ulteriori informazioni, consulta [Conversione di schemi di data warehouse in Amazon Redshift](#).
3. Nel riquadro a destra, scegli lo schema convertito.
4. Apri il menu contestuale (tasto destro del mouse) per l'elemento dello schema e quindi scegli Apply to database (Applica al database).
5. AWS SCT genera pacchetti di estensione con i componenti richiesti e aggiunge lo `aws_<database_engine_name>_ext` schema nell'albero di destinazione. Successivamente, AWS SCT applica il codice convertito e lo schema del pacchetto di estensione al data warehouse di destinazione.

Quando utilizzi una combinazione di Amazon Redshift e AWS Glue come piattaforma di database di destinazione, AWS SCT aggiunge uno schema aggiuntivo nel pacchetto di estensione.

Utilizzo delle AWS Lambda funzioni del pacchetto di AWS SCT estensione

AWS SCT fornisce un pacchetto di estensione che contiene funzioni Lambda per e-mail, pianificazione dei processi e altre funzionalità per i database ospitati su Amazon EC2.

Utilizzo di AWS Lambda funzioni per emulare la funzionalità del database

In alcuni casi, le funzionalità del database non possono essere convertite in funzionalità Amazon RDS equivalenti. Ad esempio, Oracle invia chiamate e-mail che utilizzano UTL_SMTP e Microsoft SQL Server può utilizzare un pianificatore di processi. Se ospiti e gestisci autonomamente un database su Amazon EC2, puoi emulare queste funzionalità AWS sostituendole con i servizi.

La procedura guidata AWS SCT del pacchetto di estensione consente di installare, creare e configurare le funzioni Lambda per emulare la posta elettronica, la pianificazione dei processi e altre funzionalità.

Applicazione del pacchetto di estensione per supportare le funzioni Lambda

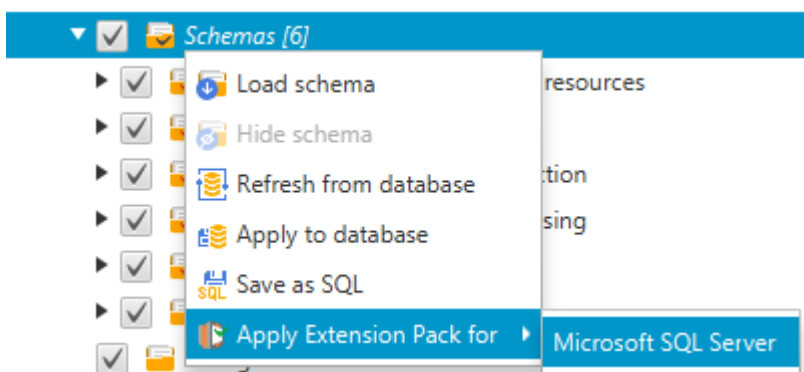
Puoi applicare il pacchetto di estensione per supportare le funzioni Lambda utilizzando la procedura guidata del pacchetto di estensione o quando applichi il codice convertito al database di destinazione.

⚠ Important

Le funzionalità AWS di emulazione del servizio sono supportate solo per i database installati e gestiti automaticamente su Amazon EC2. Non installare le funzionalità di emulazione del servizio se il database di destinazione si trova su un'istanza database Amazon RDS.

Per applicare il pacchetto di estensione utilizzando la procedura guidata per il pacchetto di estensione

1. Nell'albero del database di destinazione AWS Schema Conversion Tool, apri il menu contestuale (fai clic con il pulsante destro del mouse), scegli **Applica il pacchetto di estensione per**, quindi scegli la piattaforma di database di origine.



Viene visualizzata la procedura guidata del pacchetto di estensione.

2. Leggi la pagina **Welcome** (Benvenuti) e scegli **Next** (Avanti).
3. Nella pagina **AWS delle impostazioni del profilo**, procedi come segue:
 - Se stai reinstallando solo lo schema del pacchetto di estensione, scegli **Skip this step for now** (Salta questa fase per ora) e seleziona **Next** (Avanti).
 - Se stai installando AWS servizi, fornisci le credenziali per connetterti al tuo Account AWS. Puoi usare AWS CLI le tue credenziali se le hai installate AWS CLI . Puoi anche utilizzare le credenziali che hai precedentemente archiviato in un profilo nelle impostazioni globali dell'applicazione e associate al progetto. Se necessario scegli **Navigate to Project Settings** (Passare alle impostazioni di progetto) per associare un diverso profilo al progetto. Se necessario, scegli **Global Settings** (Impostazioni globali) per creare un nuovo profilo. Per ulteriori informazioni, consulta [Archiviazione dei profiliAWS di servizio inAWS SCT](#).
4. Nella pagina **Email Sending Service** (Servizio di invio e-mail), segui questi passaggi:

- Se stai reinstallando solo lo schema del pacchetto di estensione, scegli Skip this step for now (Salta questa fase per ora) e seleziona Next (Avanti).
 - Se stai installando AWS servizi e disponi di una funzione Lambda esistente, puoi fornirla. In caso contrario, la procedura guidata la crea automaticamente. Al termine, seleziona Next (Avanti).
5. Nella pagina Job Emulation Service (Servizio di emulazione processi), segui questi passaggi:
- Se stai reinstallando solo lo schema del pacchetto di estensione, scegli Skip this step for now (Salta questa fase per ora) e seleziona Next (Avanti).
 - Se stai installando AWS servizi e disponi di una funzione Lambda esistente, puoi fornirla. In caso contrario, la procedura guidata la crea automaticamente. Al termine, seleziona Next (Avanti).
6. Nella pagina di emulazione della funzione, scegli Crea pacchetto di estensione. I messaggi vengono visualizzati con lo stato delle operazioni del pacchetto di estensione.
- Al termine, seleziona Finish (Fine).

Note

Per aggiornare un pacchetto di estensione e sovrascrivere i vecchi componenti del pacchetto di estensione, assicurati di utilizzare la versione più recente di AWS SCT. Per ulteriori informazioni, consulta [Installazione, verifica e aggiornamento AWS SCT](#).

Configurazione delle funzioni per il pacchetto di estensione AWS SCT

Il pacchetto di estensione contiene funzioni che è necessario configurare prima dell'uso. La costante `CONVERSION_LANG` definisce il linguaggio utilizzato dal service pack. Le funzioni sono disponibili per l'inglese e il tedesco.

Per impostare la lingua su inglese o tedesco, apportate la seguente modifica al codice della funzione. Trovate la seguente dichiarazione di costante:

```
CONVERSION_LANG CONSTANT VARCHAR := '';
```

CONVERSION_LANG Per impostare l'inglese, modifica la riga nel modo seguente:

```
CONVERSION_LANG CONSTANT VARCHAR := 'English';
```

CONVERSION_LANG Per impostare l'inglese, modifica la riga nel modo seguente:

```
CONVERSION_LANG CONSTANT VARCHAR := 'Deutsch';
```

Imposta questa impostazione per le seguenti funzioni:

- `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`

Best practice per AWS SCT

Trova informazioni sulle best practice e opzioni per l'utilizzo di AWS Schema Conversion Tool (AWS SCT).

Configurazione della memoria aggiuntiva

Per convertire schemi di database di grandi dimensioni, ad esempio un database con 3.500 procedure memorizzate, è possibile configurare la quantità di memoria disponibile per AWS Schema Conversion Tool.

Per modificare la quantità di memoria che AWS SCT consuma

1. Nel menu Impostazioni, scegli Impostazioni globali, quindi scegli Opzioni JVM.
2. Scegli Modifica file di configurazione e scegli l'editor di testo per aprire il file di configurazione.
3. Modifica la sezione `JavaOptions` per impostare la quantità di memoria minima e massima disponibile. L'esempio seguente imposta il minimo a quattro GB e il massimo a 40 GB.

```
[JavaOptions]
-Xmx40960M
-Xms4096M
```

Ti consigliamo di impostare la memoria minima su almeno quattro GB di memoria minima disponibile su almeno quattro GB.

4. Salvate il file di configurazione, scegliete OK e riavviate AWS SCT per applicare le modifiche.

Configurazione della cartella di progetto predefinita

AWS SCT utilizza la cartella del progetto per archiviare i file del progetto, salvare i rapporti di valutazione e archiviare il codice convertito. Per impostazione predefinita, AWS SCT memorizza tutti i file nella cartella dell'applicazione. È possibile specificare un'altra cartella come cartella di progetto predefinita.

Per modificare la cartella di progetto predefinita

1. Nel menu Impostazioni, scegli Impostazioni globali, quindi scegli Percorso del file.

2. In Percorso predefinito del file di progetto, immettete il percorso della cartella di progetto predefinita.
3. Scegli Apply (Applica), quindi OK.

Incremento della velocità di migrazione dei dati dei dati

Per migrare set di dati di grandi dimensioni, ad esempio un set di tabelle con più di 1 TB di dati, potresti voler aumentare la velocità di migrazione. Quando si utilizzano agenti di estrazione dei dati, la velocità delle migrazioni dei dati dipende da vari fattori. Questi fattori includono il numero di sezioni nel cluster Amazon Redshift di destinazione, la dimensione di un file di blocco nell'attività di migrazione, la RAM disponibile sul PC in cui vengono eseguiti gli agenti di estrazione dei dati e così via.

Per aumentare la velocità di migrazione dei dati, consigliamo di eseguire diverse sessioni di migrazione di test con piccoli set di dati di produzione. Inoltre, ti consigliamo di eseguire gli agenti di estrazione dei dati su un PC con un SSD di almeno 500 GB di dimensione. Durante queste sessioni di test, modifica diversi parametri di migrazione e monitora l'utilizzo del disco per scoprire la configurazione che garantisce la massima velocità di migrazione dei dati. Quindi, utilizza questa configurazione per migrare l'intero set di dati.

Aumento delle informazioni di registrazione

È possibile aumentare le informazioni di registrazione prodotte AWS SCT durante la conversione dei database, degli script e del codice SQL delle applicazioni. Sebbene l'aumento delle informazioni di registrazione possa rallentare la conversione, le modifiche possono aiutarti a fornire informazioni affidabili a AWS Support in caso di errori.

AWS SCT archivia i log nell'ambiente locale. Puoi visualizzare questi file di registro e condividerli con l'AWS Support o AWS SCT gli sviluppatori per la risoluzione dei problemi.

Per modificare le impostazioni di registrazione

1. Nel menu Impostazioni, scegli Impostazioni globali, quindi scegli Registrazione.
2. Per Percorso della cartella di registro, inserisci la cartella in cui archiviare i registri dall'interfaccia utente.
3. Per il percorso della cartella di registro della console, inserisci la cartella in cui archiviare i registri dell'interfaccia a riga di AWS SCT comando (CLI).

4. In Dimensione massima del file di registro (MB), immettere la dimensione, in MB, di un singolo file di registro. Quando il file raggiunge questo limite,AWS SCT crea un nuovo file di registro.
5. In Numero massimo di file di registro, immettere il numero di file di registro da archiviare. Quando il numero di file di registro nella cartella raggiunge questo limite,AWS SCT elimina il file di registro più vecchio.
6. Per il percorso di download del registro degli estrattori, inserisci la cartella in cui archiviare i registri degli agenti di estrazione dei dati.
7. Per il percorso del registro dell'estrattore Cassandra, inserisci la cartella in cui archiviare i registri degli agenti di estrazione dei dati di Apache Cassandra.
8. Seleziona Richiedi un percorso prima del caricamento per assicurarti cheAWS SCT chieda dove archiviare i log ogni volta che usi agenti di estrazione dei dati.
9. Per la modalità Debug, scegli True. Utilizza questa opzione per registrare informazioni aggiuntive quandoAWS SCT i registri standard non includono problemi.
10. Scegli i moduli applicativi chiave per aumentare le informazioni di registrazione. È possibile aumentare le informazioni di registrazione per i seguenti moduli applicativi:
 - Generale
 - Loader
 - Analizzatore
 - stampante
 - Risolutore
 - Telemetria
 - Convertitore
 - Mappatura dei tipi
 - Interfaccia utente
 - Controller
 - Confronta lo schema
 - Clonazione del data center
 - Analizzatore di applicazioni

Per ciascuno dei seguenti livelli di destinazione dell'applicazione, scegli uno dei seguenti livelli di registrazione:

- **Trace:** informazioni più dettagliate.
- **Debug:** informazioni dettagliate sul flusso attraverso il sistema.
- **Informazioni:** eventi di runtime, come avvio o spegnimento.
- **Avviso:** utilizzo di API obsolete, scarso utilizzo delle API, altre situazioni di runtime indesiderate o impreviste.
- **Errore:** errori di runtime o condizioni impreviste.
- **Critico:** errori che portano alla chiusura dell'applicazione.
- **Obbligatorio:** il livello di errore più elevato possibile.

Per impostazione predefinita, dopo aver attivato la modalità Debug, AWS SCT imposta il livello di registrazione delle informazioni per tutti i moduli dell'applicazione.

Ad esempio, per risolvere le principali aree problematiche durante la conversione, imposta Parser, Type mapping e User interface su Trace.

Se le informazioni diventano troppo dettagliate per il file system in cui vengono trasmessi i registri, passa a una posizione con spazio sufficiente per acquisire i registri.

Per trasmettere i log a AWS Support, vai alla directory in cui sono archiviati i log e comprimi tutti i file in un unico file.zip gestibile. Quindi carica il file.zip con la tua richiesta di assistenza. Quando l'analisi iniziale è completata e lo sviluppo in corso riprende, ripristina la modalità Debug su false per eliminare la registrazione dettagliata. Quindi aumenta la velocità di conversione.

Tip

Per gestire le dimensioni dei registri e semplificare la segnalazione dei problemi, rimuovi i registri o spostali in un'altra posizione dopo una conversione riuscita. Questa operazione garantisce che solo gli errori e le informazioni pertinenti vengano trasmessi a AWS Support e impedisce il riempimento del file di registro.

Risoluzione dei problemi AWS SCT

Di seguito, troverai informazioni sui problemi relativi alla risoluzione dei problemi con AWS Schema Conversion Tool (AWS SCT).

Impossibile caricare oggetti da un database di origine Oracle

Quando tenti di caricare schemi da un database Oracle, è possibile che si verifichi uno dei seguenti errori.

```
Cannot load objects tree.
```

```
ORA-00942: table or view does not exist
```

Questi errori si verificano perché l'utente il cui ID è utilizzato per la connessione al database Oracle non dispone di autorizzazioni sufficienti per leggere lo schema, secondo quanto stabilito da AWS SCT.

Puoi risolvere il problema concedendo all'utente l'`select_catalog_role` autorizzazione insieme all'autorizzazione per qualsiasi dizionario nel database. Queste autorizzazioni forniscono l'accesso in sola lettura alle viste e alle tabelle di sistema richiesto da AWS SCT. L'esempio seguente crea un ID utente denominato `min_privs` e concede all'utente con questo ID le autorizzazioni minime necessarie per la conversione dello schema da un database di origine 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;
```

Messaggio di avviso del rapporto di valutazione

Per valutare la complessità della conversione in un altro motore di database, è necessario accedere agli oggetti nel database di origine. Quando si verificano problemi AWS SCT durante la scansione e non è possibile eseguire una valutazione, viene emesso un messaggio di avviso. Questo messaggio indica che la percentuale di conversione complessiva è ridotta. Di seguito sono riportati i motivi per cui AWS SCT potrebbero verificarsi problemi durante la scansione:

- L'utente del database non ha accesso a tutti gli oggetti necessari. Per ulteriori informazioni sulle autorizzazioni e i privilegi di sicurezza AWS SCT richiesti per il database, consulta [Origini per AWS SCT](#) la sezione appropriata sul database di origine in questa guida.
- Un oggetto citato nello schema non esiste più nel database. Per risolvere il problema, puoi connetterti con le autorizzazioni SYSDBA e verificare se l'oggetto è presente nel database.
- SCT sta cercando di valutare un oggetto crittografato.

AWS SCT Riferimento CLI

Questa sezione descrive come iniziare a usare l'interfaccia a riga di AWS SCT comando (CLI). Inoltre, questa sezione fornisce informazioni sui comandi da tastiera e sulle modalità di utilizzo. Per un riferimento completo dei comandi AWS SCT CLI, vedere. [Materiale di riferimento](#)

Argomenti

- [Prerequisiti per l'utilizzo dell'interfaccia a AWS SCT riga di comando](#)
- [AWS SCT modalità interattiva CLI](#)
- [Ottenere AWS SCT scenari CLI](#)
- [Modifica degli AWS SCT scenari CLI](#)
- [AWS SCT modalità script CLI](#)
- [AWS SCT Materiale di riferimento CLI](#)

Prerequisiti per l'utilizzo dell'interfaccia a AWS SCT riga di comando

Scarica e installa l'ultima versione di Amazon Corretto 11. Per ulteriori informazioni, consulta [Download per Amazon Corretto 11](#) nella Guida per l'utente di Amazon Corretto 11.

Scarica e installa la versione più recente di AWS SCT. Per ulteriori informazioni, consulta [Installazione AWS SCT](#).

AWS SCT modalità interattiva CLI

È possibile utilizzare l'interfaccia a riga di AWS SCT comando in modalità interattiva. In questa modalità, i comandi vengono immessi nella console uno per uno. È possibile utilizzare questa modalità interattiva per saperne di più sui comandi CLI o scaricare gli scenari CLI più utilizzati.

Per convertire lo schema del database di origine AWS SCT, esegui un'operazione di sequenza: crea un nuovo progetto, connettiti ai database di origine e di destinazione, crea regole di mappatura e converti gli oggetti del database. Poiché questo flusso di lavoro può essere complesso, consigliamo di utilizzare gli script in modalità AWS SCT CLI. Per ulteriori informazioni, consulta [Modalità script](#).

È possibile eseguire i comandi AWS SCT CLI dalla app cartella del percorso di AWS SCT installazione. In Windows, il percorso di installazione predefinito è C:\Program

Files\AWS Schema Conversion Tool\ . Assicurati che questa cartella includa il AWSSchemaConversionToolBatch.jar file.

Per accedere alla modalità interattiva AWS SCT CLI, utilizzare il comando seguente dopo aver completato i prerequisiti.

```
java -jar AWSSchemaConversionToolBatch.jar -type interactive
```

Ora puoi eseguire i comandi AWS SCT CLI. Assicurati di terminare i comandi / in una nuova riga. Inoltre, assicuratevi di utilizzare le virgolette singole diritte (') prima e dopo i valori dei parametri del comando.

Note

Se il comando precedente viene restituito `Unexpected error`, provate quanto segue:

```
java -Djdk.jar.maxSignatureFileSize=20000000 -jar  
AWSSchemaConversionToolBatch.jar
```

Per visualizzare l'elenco dei comandi disponibili in modalità interattiva AWS SCT CLI, esegui il comando seguente.

```
help  
/
```

Per visualizzare informazioni su un comando AWS SCT CLI, utilizzare il comando seguente.

```
help -command: 'command_name'  
/
```

Nell'esempio precedente, sostituite *command_name* con il nome di un comando.

Per visualizzare informazioni sui parametri di un comando AWS SCT CLI, utilizzare il comando seguente.

```
help -command: 'command_name' -parameters: 'parameters_list'  
/
```

Nell'esempio precedente, sostituite *command_name* con *il nome* di un comando. Quindi, sostituite *parameters_list* con *un elenco* di nomi di parametri separati da una virgola.

Per eseguire uno script da un file in modalità interattiva AWS SCT CLI, utilizzare il comando seguente.

```
ExecuteFile -file: 'file_path'  
/
```

Nell'esempio precedente, sostituite *file_path* con il percorso del file con uno script. Assicurati che il file abbia un'estensione. `.scts`

Per uscire dalla modalità interattiva AWS SCT CLI, esegui il `quit` comando.

Esempi

L'esempio seguente visualizza le informazioni sul `Convert` comando.

```
help -command: 'Convert'  
/
```

L'esempio seguente visualizza le informazioni su due parametri del `Convert` comando.

```
help -command: 'Convert' -parameters: 'filter, treePath'  
/
```

Ottenere AWS SCT scenari CLI

Per ottenere gli AWS SCT scenari più utilizzati, è possibile utilizzare il `GetCliScenario` comando. È possibile eseguire questo comando in modalità interattiva, quindi modificare i modelli scaricati. Usa i file modificati in modalità script.

Il `GetCliScenario` comando salva il modello selezionato o tutti i modelli disponibili nella directory specificata. Il modello contiene il set completo di comandi per eseguire uno script. Assicurati di modificare i percorsi dei file, le credenziali del database, i nomi degli oggetti e altri dati in questi modelli. Inoltre, assicuratevi di rimuovere i comandi che non utilizzate e di aggiungere nuovi comandi allo script dove necessario.

Per eseguire il `GetCliScenario` comando, completa i prerequisiti e accedi alla modalità AWS SCT interattiva CLI. Per ulteriori informazioni, consulta [Modalità interattiva](#).

Quindi, utilizzate la seguente sintassi per eseguire il GetCliScenario comando e ottenere gli scenari. AWS SCT

```
GetCliScenario -type: 'template_type' -directory: 'file_path'  
/
```

Nell'esempio precedente, sostituite *template_type* con uno dei tipi di modello della tabella seguente. Quindi, sostituisci *file_path* con il percorso della cartella in cui desideri scaricare gli script. Assicurati che AWS SCT possa accedere a questa cartella senza richiedere i diritti di amministratore. Inoltre, assicuratevi di utilizzare le virgolette singole diritte (') prima e dopo i valori dei parametri del comando.

Per scaricare tutti i modelli AWS SCT CLI, esegui il comando precedente senza l'opzione. -type

La tabella seguente include i tipi di modelli AWS SCT CLI che è possibile scaricare. Per ogni modello, la tabella include il nome del file e la descrizione delle operazioni che è possibile eseguire utilizzando lo script.

Tipo di modello	Nome file	Descrizione
BTEQ ScriptConversion	BTEQScriptConversionTemplate.scts	Converte Teradata Basic Teradata Query (BTEQ) FastExport e FastLoad gli script in Amazon Redshift RSQL. MultiLoad Per ulteriori informazioni, consulta Conversione dei processi ETL.
ConversionApply	ConversionTemplate.scts	Converte gli schemi del database di origine e applica il codice convertito al database di destinazione. Facoltativamente, salva il codice convertito come script SQL e salva il rapporto di valutazione. Per ulteriori informazioni, consulta Conversione di schemi di database.

Tipo di modello	Nome file	Descrizione
GenericAppConversion	GenericApplication ConversionTemplate .scts	Converte il codice SQL incorporato nelle applicazioni con il convertitore di AWS SCT applicazioni generico. Per ulteriori informazioni, consulta Conversione del codice SQL nelle tue applicazioni .
HadoopMigration	HadoopMigrationTemplate.scts	Esegue la migrazione del cluster Hadoop locale su Amazon EMR. Per ulteriori informazioni, consulta Utilizzo di Apache Hadoop come fonte per AWS SCT .
HadoopResumeMigration	HadoopResumeMigrationTemplate.scts	Riprende una migrazione interrotta del cluster Hadoop locale su Amazon EMR. Per ulteriori informazioni, consulta Utilizzo di Apache Hadoop come fonte per AWS SCT .
Informatica	InformaticaConversionTemplate.scts	Converte il codice SQL incorporato negli script di estrazione, trasformazione e caricamento (ETL) di Informatica. Configura le connessioni ai database di origine e di destinazione negli script ETL e salva gli script convertiti dopo la conversione. Per ulteriori informazioni, consulta Conversione di script ETL di Informatica .

Tipo di modello	Nome file	Descrizione
LanguageSpecificAppConversion	LanguageSpecificAppConversionTemplate.scts	Converte il codice SQL incorporato nelle applicazioni C#, C++, Java e Pro*C con il convertitore di applicazioni. AWS SCT Per ulteriori informazioni, consulta Conversione del codice SQL di un'applicazione .
OozieConversion	OozieConversionTemplate.scts	Converte i flussi di lavoro di Apache Oozie in. AWS Step Functions Per ulteriori informazioni, consulta Utilizzo di Apache Oozie come fonte per AWS SCT .
RedshiftAgent	DWHDataMigrationTemplate.scts	Converte gli schemi di data warehouse di origine e applica il codice convertito al database Amazon Redshift di destinazione. Quindi registra un agente di estrazione dei dati, crea e avvia un'attività di migrazione dei dati. Per ulteriori informazioni, consulta Migrazione da un data warehouse ad Amazon Redshift .

Tipo di modello	Nome file	Descrizione
ReportCreation	ReportCreationTemplate.scts	Crea un rapporto sulla migrazione del database per diversi schemi di database di origine. Quindi salva questo rapporto come file CSV o PDF. Per ulteriori informazioni, consulta Rapporti di valutazione della migrazione .
SQL ScriptConversion	SQLScriptConversionTemplate.scts	Converte gli script SQL*Plus o TSQL in PL/SQL e salva gli script convertiti. Inoltre, salva un rapporto di valutazione.

Dopo aver scaricato il modello AWS SCT CLI, utilizza l'editor di testo per configurare lo script da eseguire sui database di origine e di destinazione. Quindi, usa la modalità script AWS SCT CLI per eseguire lo script. Per ulteriori informazioni, consulta [AWS SCT modalità script CLI](#).

Esempi

L'esempio seguente scarica tutti i modelli nella C:\SCT\Templates cartella.

```
GetCliScenario -directory: 'C:\SCT\Templates'  
/
```

L'esempio seguente scarica il modello per l'ConversionApplyoperazione nella C:\SCT\Templates cartella.

```
GetCliScenario -type: 'ConversionApply' -directory: 'C:\SCT\Templates'  
/
```

Modifica degli AWS SCT scenari CLI

Dopo aver scaricato i modelli di scenario, configurali per ottenere script funzionanti da eseguire sui tuoi database.

Per tutti i modelli, assicurati di fornire il percorso dei driver per i database di origine e di destinazione. Per ulteriori informazioni, consulta [Scaricamento dei driver di database richiesti](#).

Assicurati di includere le credenziali del database per i database di origine e di destinazione. Inoltre, assicuratevi di impostare delle regole di mappatura per descrivere una coppia sorgente-destinazione per il vostro progetto di conversione. Per ulteriori informazioni, consulta [Creazione di regole di mappatura](#).

Quindi, configura l'ambito delle operazioni da eseguire. È possibile rimuovere i comandi che non si utilizzano o aggiungere nuovi comandi allo script.

Ad esempio, supponiamo di voler convertire tutti gli schemi del database Oracle di origine in PostgreSQL. Quindi intendi salvare il rapporto di valutazione della migrazione del database in formato PDF e applicare il codice convertito al database di destinazione. In questo caso, puoi utilizzare il modello per l'ConversionApplyoperazione. Utilizza la procedura seguente per modificare il modello AWS SCT CLI.

Per modificare il modello AWS SCT CLI per l'operazione **ConversionApply**

1. Apri quello `ConversionTemplate.scts` che hai scaricato. Per ulteriori informazioni, consulta [Esempi](#).
2. Rimuovi `CreateFilter`, converti le operazioni `-filter`, `ApplyToTarget -filter`, `SaveTargetSQL`, `SaveTargetSQLbyStatement` e `SaveReportCSV` dallo script.
3. Per `oracle_driver_file` nell'operazione, inserisci il percorso del tuo driver Oracle. `SetGlobalSettings` Quindi, per `postgresql_driver_file`, inserisci il percorso del tuo driver PostgreSQL.

Se utilizzi altri motori di database, usa i nomi appropriati per le impostazioni. Per un elenco completo delle impostazioni globali che è possibile impostare `SetGlobalSettings` durante l'operazione, vedere Matrice delle impostazioni globali in [Materiale di riferimento](#).

4. (Facoltativo) Per `CreateProject`, inserite il nome del progetto e la posizione del file di progetto locale. Se scegli di procedere con i valori predefiniti, assicurati che sia AWS SCT possibile creare file nella `C:\temp` cartella senza richiedere i diritti di amministratore.
5. Per `AddSource`, inserisci l'indirizzo IP del server del database di origine. Inoltre, inserisci il nome utente, la password e la porta per connetterti al server del database di origine.
6. Per `AddTarget`, inserisci l'indirizzo IP del server di database di destinazione. Inoltre, inserisci il nome utente, la password e la porta per connetterti al server del database di destinazione.

7. (Facoltativo) Per AddServerMapping, inserisci gli oggetti del database di origine e di destinazione che desideri aggiungere a una regola di mappatura. È possibile utilizzare `targetTreePath` i parametri `sourceTreePath` e per specificare il percorso degli oggetti del database. Facoltativamente, è possibile `targetNamePath` utilizzare `sourceNamePath` e specificare i nomi degli oggetti del database. Per ulteriori informazioni, vedere Comandi di mappatura del server in. [Materiale di riferimento](#)

I valori predefiniti dell'AddServerMappingoperazione mappano tutti gli schemi di origine con il database di destinazione.

8. Salvate il file e quindi utilizzate la modalità script per eseguirlo. Per ulteriori informazioni, consulta [Modalità script](#).

AWS SCT modalità script CLI

Dopo aver creato uno script AWS SCT CLI o modificato un modello, puoi eseguirlo con il RunSCTBatch comando. Assicurati di salvare il file con lo script CLI come estensione. `.scts`

È possibile eseguire gli script AWS SCT CLI dalla app cartella del percorso di installazione. AWS SCT In Windows, il percorso di installazione predefinito è. `C:\Program Files\AWS Schema Conversion Tool\` Assicurati che questa cartella includa il `RunSCTBatch.sh` file `RunSCTBatch.cmd` or. Inoltre, questa cartella dovrebbe includere il `AWSSchemaConversionToolBatch.jar` file.

In alternativa, è possibile aggiungere il percorso del RunSCTBatch file nella variabile di PATH ambiente del sistema operativo. Dopo aver aggiornato la variabile di PATH ambiente, puoi eseguire gli script AWS SCT CLI da qualsiasi cartella.

Per eseguire uno script AWS SCT CLI, utilizzate il seguente comando in Windows.

```
RunSCTBatch.cmd --pathtoscts "file_path"
```

Nell'esempio precedente, sostituite *file_path* con il percorso del file con uno script.

Per eseguire uno script AWS SCT CLI, utilizzare il seguente comando in Linux.

```
RunSCTBatch.sh --pathtoscts "file_path"
```

Nell'esempio precedente, sostituite *file_path* con il percorso del file con uno script.

È possibile fornire parametri opzionali in questo comando, come le credenziali del database, il livello di dettagli nell'output della console e altri. Per ulteriori informazioni, scaricate il riferimento all'interfaccia della riga di AWS SCT comando all'indirizzo [Materiale di riferimento](#).

Esempi

L'esempio seguente esegue lo `ConversionTemplate.scts` script nella `C:\SCT\Templates` cartella. È possibile utilizzare questo esempio in Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\SCT\Templates\ConversionTemplate.scts"
```

L'esempio seguente esegue lo `ConversionTemplate.scts` script nella `/home/user/SCT/Templates` directory. È possibile utilizzare questo esempio in Linux.

```
RunSCTBatch.sh --pathtoscts "/home/user/SCT/Templates/ConversionTemplate.scts"
```

AWS SCT Materiale di riferimento CLI

[È possibile trovare materiale di riferimento sull'interfaccia a riga di AWS Schema Conversion Tool comando \(CLI\) nella seguente guida: CLI AWS Schema Conversion Tool Reference.](#)

Note di rilascio per AWS SCT

Questa sezione contiene le note di rilascio per AWS SCT, a partire dalla versione 1.0.640.

Note di rilascio per la build 675 AWS SCT

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
Cassandra	DynamoDB	Risolto un bug a causa del quale l'installazione di Cassandra non funzionava sul Datacenter di destinazione.	No
DB2 LUW	PostgreSQL	SQL DINAMICO: istruzione PREPARE: risoluzione e conversione senza SQL dinamico.	No
DB2 LUW	PostgreSQL	Aggiunto il supporto per SPECIAL REGISTER.	No
DB2 LUW	PostgreSQL	Aggiornamento del pacchetto di estensione	No
Hadoop	Amazon EMR	È stato aggiunto il supporto per la connessione a un cluster Hadoop tramite il protocollo rsa-sha2.	No
Microsoft SQL Server	Amazon Redshift	Risolto il problema per cui il driver JDBC forzava TLS nonostante non fosse configurato.	No
Netezza	Amazon Redshift	È stato aggiunto il supporto per la conversione delle viste materializzate.	No
Oracle	Amazon Redshift	È stato aggiunto il supporto per le query ricorsive in Amazon Redshift.	Sì

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
Oracle	PostgreSQL, Aurora PostgreSQL	Correzione della conversione errata del tipo di dati NUMBER.	Sì
Oracle	Amazon Redshift	Migrazione dei dati. Partizionamento automatico Oracle. È stata aggiunta la data di scadenza per il valore dei frammenti di tabella. Il tempo di scadenza è 72 ore. Quando si verifica la scadenza, i frammenti di dati vengono ricostruiti quando viene creata un'attività di migrazione dei dati.	No
Oracle	Amazon Redshift	SCT Data Extractor: è stato modificato l'approccio al caricamento dei dati su Amazon Redshift. Per impostazione predefinita, l'estrattore non crea tabelle organizzate. Invece, dopo che tutti i file di dati sono nel bucket Amazon S3, l'estrattore li copia nella tabella di destinazione utilizzando un singolo comando COPY.	No
Oracle	Amazon Redshift	È stata aggiunta la migrazione del tipo di dati RAW nella colonna VARBYTE.	No
Oracle	PostgreSQL, Aurora PostgreSQL	Conversione multiversione	No
Oracle	PostgreSQL	È stato aggiunto il supporto per MERGE Statement in PostgreSQL 15.x.	Sì

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
Oracle	PostgreSQL	È stato aggiunto il supporto per le nuove funzioni di espressione regolare in PostgreSQL 15.x.	Sì
Oracle	PostgreSQL, Aurora PostgreSQL	L'istruzione ON CONFLICT DO UPDATE viene convertita senza alias esclusi.	Sì
Teradata	Amazon Redshift	È stato aggiunto il supporto di conversione per le funzioni LEAD/LAG.	No
Teradata	Amazon Redshift	Casting avanzato dei tipi di dati con indicazione esplicita del formato dei dati.	No
Teradata	Amazon Redshift	Conversione migliorata della clausola AT 'TIME ZONE' nelle espressioni time/timestamp.	No
Teradata	Amazon Redshift	AI-9996 durante le procedure di conversione con istruzioni MERGE.	No

Note di rilascio per Build 674 AWS SCT

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
Tutti	Tutti	Varie correzioni di bug e miglioramenti delle prestazioni	Parziale (solo per le coppie di origine e destinazione supportate)

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
SQL di Azure/Microsoft SQL Server	Amazon Redshift	È stato rimosso il messaggio «AI 18066: Impossibile convertire il nome dello schema» che ingannava gli utenti durante la valutazione/conversione dello schema	No
SQL di Azure/Microsoft SQL Server	Amazon RDS per MySQL//Amazon Aurora MySQL	Conversione errata della procedura senza assegnazione di un codice di reso	Parziale (Schema Conversion attualmente non supporta Azure SQL come sorgente)
SQL di Azure/Microsoft SQL Server	Amazon RDS per PostgreSQL/Amazon Aurora PostgreSQL	Risolto il problema con AI9997 per alcuni casi di conversione della clausola FOR XML PATH	Parziale (Schema Conversion attualmente non supporta Azure SQL come origine)

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
SQL di Azure/Microsoft SQL Server	Amazon RDS per PostgreSQL/Amazon Aurora PostgreSQL	Il valore è arrotondato alla scala originale nel corpo della procedura/funzione	Parziale (Schema Conversion attualmente non supporta Azure SQL come origine)
SQL di Azure/Microsoft SQL Server	Amazon RDS per PostgreSQL/Amazon Aurora PostgreSQL	Vari miglioramenti alla conversione delle istruzioni EXECUTE	Parziale (Schema Conversion attualmente non supporta Azure SQL come sorgente)
SQL di Azure/Microsoft SQL Server/Azure Synapse	Amazon Redshift	Conversione migliorata delle seguenti istruzioni e modalità: <ul style="list-style-type: none"> • BLOCCO DI ECCEZIONI • AUTOCOMMIT • NONATOMIC • SET DI RAGGRUPPAMENTO • CUBE • ROLLUP 	No

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
DB2 LUW	Amazon RDS per PostgreSQL / Amazon Aurora PostgreSQL	Diverse correzioni al caricamento dei metadati nelle query sql	No
DB2 LUW	Amazon RDS per PostgreSQL / Amazon Aurora PostgreSQL	Al 9996 non è previsto sui trigger	No
DB2 z/OS	Amazon RDS per PostgreSQL / Amazon Aurora PostgreSQL	Funzione analitica ROWNUMBER	No

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
DB2 z/OS	Amazon RDS per PostgreSQL / Amazon Aurora PostgreSQL	Supporto costante per stringhe esadecimali	No
DB2 z/OS	Amazon RDS per PostgreSQL / Amazon Aurora PostgreSQL	Diverse correzioni al caricamento dei metadati nelle query sql.	No
DB2 z/OS	Amazon RDS per PostgreSQL / Amazon Aurora PostgreSQL	Supporto per riferimenti di sequenza NEXT VALUE FOR	No

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
DB2 z/OS	Amazon RDS per PostgreSQL/ Amazon Aurora PostgreSQL	Supporto dell'opzione DB2_NUMBER_ROWS per l'istruzione GET DIAGNOSTICS	No
DB2 z/OS	Amazon RDS per PostgreSQL/ Amazon Aurora PostgreSQL	GET DIAGNOSTICS Dichiarazioni multiple	No
DB2 z/OS	Amazon RDS per PostgreSQL/ Amazon Aurora PostgreSQL	Sono stati corretti alcuni bug nella conversione delle istruzioni FOR.	No

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
Oracle	Amazon RDS per MySQL// Amazon Aurora MySQL	Risolto un bug che si verificava quando il nodo dei parametri della funzione del pacchetto non era definito.	Sì
Oracle	Amazon RDS per PostgreSQL/ Amazon Aurora PostgreSQL	Sono stati corretti alcuni bug nella funzione AWS_ORACLE_EXT.NEXT_DAY del pacchetto di estensione	Sì
Oracle	Amazon RDS per PostgreSQL/ Amazon Aurora PostgreSQL	Sono stati corretti diversi bug relativi alla conversione «(+)» nei join esterni di Oracle	Sì
Oracle		Supporta l'autenticazione Kerberos	No

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
SAP ASE	Amazon RDS per PostgreSQL/ Amazon Aurora PostgreSQL	È stato corretto un bug che si verificava durante la conversione di più di un identificatore nella clausola FROM nell'istruzione UPDATE	No
SAP ASE	Amazon RDS per PostgreSQL/ Amazon Aurora PostgreSQL	Risolto un bug con la conversione di commenti e istruzioni su più righe	No
SAP ASE		Aggiunto il supporto per il parametro ENCRYPT_PASSWORD durante la connessione	No
Teradata	Amazon Redshift	Conversione migliorata della tabella VOLATILE con il nome dello schema specificato	No
Teradata	Amazon Redshift	Conversione errata della CLAUSOLA WHERE nel complesso CTE	No
Teradata	Amazon Redshift	È stato aggiunto il supporto per il tipo di dati INTERVAL durante la migrazione dei dati utilizzando SCT Data Extraction Agents.	No

Origine	Target	Novità, miglioramento o correzione	Disponibilità in AWS DMS Schema Conversion
Script Teradata BTEQ	Script RSQL di Amazon Redshift	Parametri di conversione in uscita errati nella procedura eseguita da BTEQ	No

Note di rilascio per Build 673 AWS SCT

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	Correzioni di bug generali e miglioramenti delle prestazioni
SQL di Azure/Microsoft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	È stata corretta la conversione errata delle chiamate di funzione
SQL di Azure/Microsoft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Conversione implementata della clausola FOR XML
SQL di Azure/Microsoft SQL Server	Aurora PostgreSQL/Amazon RDS	Conversione di una clausola XML con FOR un alias errato.

Origine	Target	Novità, miglioramento o correzione
	PostgreSQL	
SQL di Azure/Microsoft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Risolto un bug che impediva la conversione EXECUTE delle istruzioni che eseguono AWS SCT una stringa di caratteri con parametri di procedura.
SQL di Azure/Microsoft SQL Server	Aurora PostgreSQL/Amazon RDS PostgreSQL	Conversione migliorata delle istruzioni UPDATE con join interni.
Azure Synapse	Amazon Redshift	Risolve la conversione errata della funzione OBJECT_ID integrata.
IBM DB2 per z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	Implementata la conversione delle seguenti istruzioni e oggetti: <ul style="list-style-type: none"> • DECLARE TEMPORARY TABLE statement • DROP TABLE statement • PKe UNIQUE vincoli sulle tabelle partizionate • Funzione TIMESTAMPDIFF • Funzione TO_DATE • Funzione EBCDIC_STR • Funzione VARCHAR_FORMAT

Origine	Target	Novità, miglioramento o correzione
IBM DB2 per z/OS	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	È stato corretto un bug che si verificava quando l'indice basato sulle funzioni ignorava le funzioni dopo la conversione.
IBM DB2 per z/OS	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Risolto un bug per cui l'istruzione si chiudeva con AI 9996 dopo la conversione REPEAT
IBM DB2 per z/OS	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Risolto un bug in cui la clausola si chiudeva FINAL TABLE con 9996.
IBM DB2 per z/OS	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	LOADER Chiave di partizionamento nel vincolo dei riferimenti. AWS SCT ora può convertire le chiavi primarie e i vincoli univoci in tabelle partizionate come indici secondari.
IBM DB2 per z/OS	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Supporto per la funzione PostgreSQL.varChar_format

Origine	Target	Novità, miglioramento o correzione
IBM DB2 per z/OS	Aurora PostgreSQL/Amazon RDS PostgreSQL	Implementata la modifica delle regole di confronto <code>CreateTransformationRule</code> e i comandi <code>ModifyTransformationRule</code> SCT CLI.
Greenplum	Amazon Redshift	Risolto un bug con una chiamata errata della stored procedure dopo la conversione
Hadoop	Amazon EMR	È stato aggiunto il supporto per la connessione a un cluster Hadoop utilizzando il protocollo <code>rsa-sha2</code> .
Hadoop	Amazon EMR	È stato aggiunto il supporto per Amazon EMR con metastore diverso da Glue Hive,
Oracle	Amazon Redshift	È stato corretto un bug che causava la conversione errata della query ricorsiva in cui la colonna <code>PRIOR</code> non era presente nell'elenco <code>SELECT</code> .
Oracle	Aurora PostgreSQL/Amazon RDS PostgreSQL	Implementata la restituzione di un elemento di un array associativo
Oracle	Aurora PostgreSQL/Amazon RDS PostgreSQL	Risolto un problema imprevisto con <code>AI 9996</code> con parentesi <code>UNPIVOT</code>

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	Risolto un problema imprevisto con <code>AI 9996 UNPIVOT UNION ALL</code>
Oracle	Aurora PostgreSQL L/Amazon RDS PostgreSQL L	miglioramenti per le conversioni dei <code>Number</code> tipi di dati
Oracle	Estrattore e di dati Amazon Redshift	Supporto per il partizionamento automatico per le tabelle Oracle. Ottimizzazione per la creazione di attività di migrazione.
Teradata	Amazon Redshift	Implementare la conversione della <code>EXCEPTION BLOCK</code> dichiarazione
Teradata	Amazon Redshift	Supporto per la conversione e <code>ALL ANY</code> i <code>SOME</code> predicati in Amazon Redshift.
Teradata	Amazon Redshift	È stato aggiunto il supporto nativo per <code>QUALIFY</code> i predicati.
Teradata	Amazon Redshift	Conversione migliorata di quanto segue: <ul style="list-style-type: none"> • Interrogazioni ricorsive • <code>GROUPING SET</code> • <code>CUBE</code> • <code>ROLLUP</code> • <code>UPDATE</code> dichiarazione con join implicito

Origine	Target	Novità, miglioramento o correzione
sorgenti OLAP	Estrattore e di dati Amazon Redshift	Comandi CLI implementati per le attività Stop/Resume for Amazon Redshift Data Extractor.
Fonti OLAP	Estrattore e di dati Amazon Redshift	È stata aggiunta la possibilità di selezionare le colonne della tabella da migrare durante la configurazione dell'attività di migrazione.

Note di rilascio per AWS SCT Build 672

Origine	Target	Novità, miglioramento o correzione
Tutti	Amazon RDS per PostgreSQL	Supporto implementato della versione principale 15 di PostgreSQL come obiettivo di migrazione.
Tutti	Amazon Redshift	È stato aggiunto un nuovo <code>PrintTaskStatus</code> AWS SCT comando nell'interfaccia a riga di comando (CLI) per visualizzare lo stato dell'attività di migrazione dei dati.
Tutti	Amazon Redshift	È stato migliorato il flusso di configurazione per gli agenti di estrazione e dei dati.
Tutti	Amazon Redshift	È stato risolto un errore a causa del quale gli agenti di estrazione dei dati non visualizzavano le informazioni sulle sottoattività.
Apache Oozie	AWS Step Functions	È stata aggiunta un'opzione per salvare le definizioni delle macchine a stati come script nel codice convertito.
Database SQL di Azure	Aurora PostgreSQL	Implementata la conversione di <code>COALESCEDATEADD</code> , <code>GETDATE</code> , e <code>SUM</code> funzioni.

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	PostgreSQL	
Database SQL di Azure	Aurora PostgreSQL	È stata migliorata la conversione delle UPDATE istruzioni con OUTPUT clausole JOIN e.
Microsoft SQL Server	PostgreSQL	
Database SQL di Azure	Aurora PostgreSQL	È stato risolto un errore che si verificava durante la conversione dell'SELECT TOP 1 WITH TIESistruzione.
Microsoft SQL Server	PostgreSQL	
Database SQL di Azure	Aurora PostgreSQL	Sono stati risolti diversi problemi che si verificavano durante la conversione delle FOR XML clausole in funzioni integrate.
Microsoft SQL Server	PostgreSQL	
Greenplum	Amazon Redshift	Implementata la conversione di RAISE istruzioni GET DIAGNOSTICS e utilizzando un blocco Amazon Redshift EXCEPTION nativo.
Greenplum	Amazon Redshift	È stata migliorata la conversione delle procedure memorizzate aggiungendo il supporto di un EXCEPTION blocco nel codice convertito.

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato corretto un errore a causa del quale TO_CHAR la funzione con i modelli di formato temporale veniva convertita in modo errato.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione delle espressioni di tabella annidate.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione delle istruzioni GOTO, MERGE, REPEAT e. SIGNAL
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione delle FETCH istruzioni con BEFORE parole chiave di orientamento. AFTER
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione dei riferimenti alle tabelle. FINAL TABLE OLD TABLE

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione delle seguenti funzioni. <ul style="list-style-type: none"> • ADD_MONTHS • DAYcon parametri del tipo di dati del carattere • 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 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione di un alias di una sottoquery in clause. JOIN

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Migliorata la conversione delle funzioni. COALESCE
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione degli indici. EXPLICIT
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione dei nomi delle colonne nelle espressioni composte per risolvere un problema a causa del quale l'azione 9997 appare inaspettatamente durante la conversione.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle chiavi primarie e dei vincoli unici.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle XMLTABLE istruzioni nelle INSERT istruzioni per risolvere un problema a causa del quale l'azione 9996 appare inaspettatamente durante la conversione.

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL PostgreSQL	È stato risolto un problema a causa del quale l'azione 9996 appariva inaspettatamente durante la conversione delle funzioni con l'argomento. SUBSTR
IBM Db2 per z/OS	Aurora PostgreSQL PostgreSQL	È stato risolto un problema a causa del quale l'azione 9996 compariva inaspettatamente durante la conversione del registro speciale. CURRENT_TIMESTAMP
IBM Db2 per z/OS	Aurora PostgreSQL PostgreSQL	È stato risolto un problema a causa del quale l'azione 9996 appariva inaspettatamente durante la conversione di MERGE istruzioni, istruzioni non supportate e funzioni integrate non supportate.
Microsoft SQL Server	Tutti	È stato aggiunto il supporto di Microsoft SQL Server versione 2022 come fonte.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	È stata migliorata la conversione delle SELECT istruzioni che utilizzano operatori di concatenazione di stringhe. AWS SCT utilizza la STRING_AGG funzione nel codice convertito.
Microsoft SQL Server	Babelfish per Aurora PostgreSQL	È stato implementato il supporto della nuova versione 3.1.0 del file di configurazione delle funzionalità di Babelfish. Questo file definisce le funzionalità SQL supportate e non supportate da versioni specifiche di Babelfish.

Origine	Target	Novità, miglioramento o correzione
Netezza	Amazon Redshift	È stato risolto un problema a causa del quale gli agenti di estrazione dei dati non avviavano la migrazione dei dati dal punto CDC specificato.
Oracle	Tutti	È stato aggiornato il rapporto di valutazione per la versione 19 dei database Oracle come fonte.
Oracle	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione del DBMS_OUTPUT pacchetto aggiungendo nuove funzioni al pacchetto di AWS SCT estensione.
Oracle	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione di funzioni e procedure che utilizzano array associativi come argomenti o parametri.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle DISTINCT clausole nelle istruzioni. SELECT
Oracle	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle tabelle in cui il vincolo della chiave primaria ha lo stesso nome della tabella.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	Migliorata la conversione della RAISE_APPLICATION_ERROR procedura con il terzo parametro.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale la regola di migrazione e non modificava automaticamente il tipo di NUMERIC dati INTEGER laddove applicabile.
Oracle DW	Amazon Redshift	Supporto implementato delle CONNECT BY clausole native di Amazon Redshift nel codice convertito.
Oracle DW	Amazon Redshift	È stata migliorata la migrazione dei dati aggiungendo automaticamente una sottoattività per ogni tabella o partizione nell'ambito della migrazione. Questo approccio impedisce la perdita di dati per i dati inseriti dopo il partizionamento.
Teradata	Amazon Redshift	Implementata la conversione delle viste ricorsive.
Teradata	Amazon Redshift	È stata migliorata la conversione delle stored procedure che utilizzano BTET e ANSI delle modalità di transazione aggiungendo il supporto della modalità di AUTOCOMMIT transazione nativa di Amazon Redshift.
Teradata	Amazon Redshift	È stata migliorata la conversione delle stored procedure che utilizzano la semantica TERADATA delle transazioni aggiungendo la NONATOMIC parola chiave nel codice convertito.
Teradata	RSQL Amazon Redshift	È stato risolto un problema a causa del quale il codice convertito includeva l'ID della chiave di AWS accesso e la chiave di accesso segreta.

Note di rilascio per la AWS SCT build 671

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	È stato corretto un errore in cui AWS SCT non si disponeva delle autorizzazioni per salvare un file di progetto in Windows.
Tutti	Tutti	<p>Sono stati aggiornati i seguenti AWS SCT modelli di interfaccia a riga di comando (CLI).</p> <ul style="list-style-type: none">• BTEQ ScriptConversion• ConversionApply• HadoopMigration• HadoopResumeMigration• Informatica <p>Per ulteriori informazioni sui modelli AWS SCT CLI, consulta Ottenere scenari CLI</p>
Tutti	Amazon Redshift	È stato corretto un errore AWS SCT che impediva la creazione di un pacchetto di estensione nell'interfaccia a riga di comando (CLI).
Tutti	Amazon Redshift	È stato risolto un problema a causa del quale gli agenti di estrazione e AWS SCT dati non utilizzavano la AWS Snowball configurazione nell'interfaccia a riga di comando (CLI).
Apache Oozie	AWS Step Functions	Supporto implementato per la migrazione da Apache Oozie alla modalità interfaccia AWS Step Functions a riga di comando (CLI). Dopo aver migrato i carichi di lavoro Hadoop su Amazon EMR, ora puoi migrare il sistema di pianificazione del flusso di lavoro su Cloud AWS. Per ulteriori informazioni, consulta Convertire Apache Oozie in AWS Step Functions .
Database SQL di Azure	Aurora PostgreSQL	È stato corretto un errore del resolver che si verificava per tabelle e alias.

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	PostgreSQL	
Database SQL di Azure	Aurora PostgreSQL	Implementata la conversione delle INDEX ON clausole.
Microsoft SQL Server	PostgreSQL	
Database SQL di Azure	Aurora PostgreSQL	È stata migliorata la conversione dei seguenti oggetti per evitare azioni impreviste.
Microsoft SQL Server	PostgreSQL	<ul style="list-style-type: none"> • Dichiarazioni Batch • Elenchi di espressioni • Alias di tabella • Tabelle temporanee • Trigger • Variabili utente
Database SQL di Azure	Aurora PostgreSQL	È stato risolto un errore di analisi che si verificava per le procedure.
Microsoft SQL Server	PostgreSQL	

Origine	Target	Novità, miglioramento o correzione
Database SQL di Azure	Aurora PostgreSQL	È stato corretto un errore in cui AWS SCT venivano utilizzati nomi errati di tabelle temporanee nel codice convertito per OBJECT_ID le funzioni.
Microsoft SQL Server	PostgreSQL	
Database SQL di Azure	Aurora PostgreSQL	Sono stati risolti i problemi a causa dei quali l'azione 9996 compariva inaspettatamente durante la conversione dei seguenti elementi di codice.
Microsoft SQL Server	PostgreSQL	<ul style="list-style-type: none"> • Funzioni CONVERT • Funzioni DATEADD • DELETE istruzioni all'interno di funzioni in linea • Istruzioni IF • INSERTo UPDATE azioni su una colonna • Istruzioni RETURN • UPDATE istruzioni con domande o funzioni complesse
BigQuery	Amazon Redshift	È stato aggiunto il supporto BigQuery come fonte per il processo di valutazione multiserver. Per ulteriori informazioni, consulta Creazione di un rapporto di valutazione multiserver .
Hadoop	Amazon EMR	È stata aggiornata la versione del driver JDBC Apache Hive supportato che utilizzi per connetterti ai database di origine. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti .

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato migliorato il caricatore di metadati di origine per garantire che AWS SCT carichi oggetti del database di origine come chiavi primarie, indici impliciti e così via.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato corretto un errore del resolver che si verificava per le colonne nei cursori impliciti.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la capacità di mantenere la formattazione dei nomi di colonna, delle espressioni e delle clausole nelle istruzioni DML nel codice convertito.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione di chiavi esterne tra schemi.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione di e funzioni. LENGTH VARCHAR

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione delle istruzioni e. LABEL ON DECLARE CONDITION
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione di dichiarazioni con clausole SELECT. OPTIMIZE FOR
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle CREATE TABLE istruzioni aggiungendo valori predefiniti per tutti i tipi di dati supportati.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione degli attributi. INCREMENT BY
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle tabelle partizionate aggiungendo la possibilità di escludere le partizioni di tabella dall'ambito di conversione.

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle definizioni delle chiavi primarie con colonne. INCLUDE
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Migliorata la conversione della funzione. SUBSTRING
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle istruzioni e delle istruzioni. SET DECLARE HANDLER FOR
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione dei tipi di dati variabili.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Migliorata la conversione delle funzioni. XMLTABLE

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato migliorato il flusso di migrazione implementando il seguente ordine di applicazione degli oggetti convertiti al database di destinazione: tabelle, partizioni, indici, vincoli, chiavi esterne e trigger.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale l'azione 9996 appariva inaspettatamente durante la conversione dei commenti nel codice sorgente.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale l'azione 9997 compariva inaspettatamente durante la conversione degli alias nelle clausole. FROM
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale l'azione 9997 appariva inaspettatamente durante la conversione degli alias del cursore.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stato corretto un errore a causa del quale il codice convertito restituiva risultati diversi per le istruzioni con clausole. SELECT ORDER BY Poiché SQL Server e PostgreSQL NULL trattano i valori in modo diverso, il codice convertito ora NULLS FIRST include clausole NULLS LAST o che assicurano che il codice convertito restituisca i risultati nello stesso ordine del codice sorgente.

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale i tipi di dati nelle funzioni di tabella venivano convertiti in modo errato.
MySQL	Amazon RDS per MySQL	È stato risolto un problema a causa del quale le virgolette singole (' ') apparivano inaspettatamente tra i nomi degli oggetti del database nel codice convertito.
Oracle	Aurora PostgreSQL L PostgreSQL L	Sono state aggiunte nuove visualizzazioni al pacchetto di estension e per emulare le viste di sistema Oracle utilizzate per visualizzare informazioni su partizioni e partizioni secondarie.
Oracle	Aurora PostgreSQL L PostgreSQL L	Sono state aggiornate due funzioni nel pacchetto di estensione per aggiungere nomi di schema come argomenti nel codice convertito.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato corretto un errore AWS SCT che impediva l'utilizzo dei parametri corretti per la conversione delle applicazioni C++ dopo l'aggiornamento del codice dell'applicazione nell'interfaccia utente.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle CREATE TYPE istruzioni per evitare eccezioni impreviste.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle tabelle annidate.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un errore di analisi che si verificava per gli oggetti del pacchetto.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale i nomi degli oggetti venivano tagliati in modo AWS SCT imprevisto nel codice convertito quando la lunghezza del nome superava i 60 caratteri.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale i trigger a livello di riga per le tabelle partizionate venivano convertiti in modo errato.

Origine	Target	Novità, miglioramento o correzione
Oracle DW	Amazon Redshift	Supporto implementato del partizionamento automatico delle tabelle per la migrazione dei dati. Per accelerare la migrazione dei dati, AWS SCT può partizionare automaticamente tabelle o partizioni di grandi dimensioni in base ai valori nella pseudocolonna. ROWID Per ulteriori informazioni, consulta Utilizzo del partizionamento nativo .
Teradata	Amazon Redshift	Supporto implementato dei MERGE comandi nativi nel codice Amazon Redshift convertito. Per ulteriori informazioni sul MERGE comando in Amazon Redshift, consulta MERGE nella Amazon Redshift Database Developer Guide.
Teradata	Amazon Redshift	È stata migliorata la conversione di UPDATE istruzioni DELETE e istruzioni che non utilizzano nomi di tabella espliciti.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale NOT IN le istruzioni IN and venivano convertite in modo errato.

Note di rilascio per AWS SCT Build 670

Origine	Target	Novità, miglioramento o correzione
Database SQL di Azure	Aurora PostgreSQL	Sono stati risolti i problemi a causa dei quali l'azione 9996 compariva inaspettatamente durante la conversione dei seguenti elementi di codice.
Microsoft SQL Server	PostgreSQL	<ul style="list-style-type: none"> • CREATE INDEX dichiarazioni all'interno delle dichiarazioni INCLUDE • Istruzioni DECLARE • Istruzioni DECLARE . . . TABLE • DECLARE con valori predefiniti all'interno LOOP delle istruzioni • Istruzioni DELETE • DROP CONSTRAINT dichiarazioni all'interno ALTER TABLE delle dichiarazioni

Origine	Target	Novità, miglioramento o correzione
		<ul style="list-style-type: none"> • EXECUTE AS CALLER e REVERT • Istruzioni IIF • Elenchi di espressioni • Funzioni MONTH() • Istruzioni UPDATE • Funzioni YEAR()
Analisi di Azure Synapse	Amazon Redshift	È stato aggiunto il supporto per Azure Synapse Analytics come fonte per il processo di valutazione multiserver. Per ulteriori informazioni, consulta Creazione di un rapporto di valutazione multiserver .
Hadoop	Amazon EMR	Supporto implementato per la migrazione dei cluster Hadoop su Amazon EMR in modalità interfaccia a riga di comando (CLI). Per ulteriori informazioni, consulta Migrazione di framework per big data .
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato corretto un errore del resolver che si verificava per le tabelle e le colonne di origine.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione delle espressioni. CASE
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione dei CURRENT_DATE riferimenti in registri speciali. Un riferimento a un registro speciale in Db2 for z/OS è un riferimento a un valore fornito dal server corrente.

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata la conversione di e funzioni. DATE POSSTR
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle costanti di data/ora.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione dei valori predefiniti per le colonne dei seguenti tipi di dati:DATE,, TIME e. TIMESTAMP TIMESTAMP WITH TIME ZONE
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale l'azione 9996 compariva inaspettatamente durante la conversione delle istruzioni. SELECT INTO
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione delle funzioni. DATEDIFF

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	È stato corretto un errore in cui ISNULL le funzioni venivano convertite in NULLIF. Di conseguenza, il codice convertito ha prodotto risultati diversi rispetto al codice sorgente. Ora AWS SCT converte ISNULL le funzioni in COALESCE
Netezza	Amazon Redshift	Agenti di estrazione dati migliorati per risolvere un problema in cui lo stato di errore era impostato per le attività completate con successo.
Netezza	Amazon Redshift	È stata aggiunta la possibilità di modificare gli endpoint nelle sottoattività dopo l'avvio di una migrazione dei dati con agenti di estrazione dei dati.
Microsoft SQL Server MySQL Oracle PostgreSQL	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	È stata aggiunta la possibilità di connettersi ai database utilizzando un protocollo di indirizzi IPv6.
Oracle	Amazon RDS per Oracle	Conversione implementata del DBMS_JOB pacchetto che pianifica e gestisce i lavori nella coda dei lavori.
Oracle	Aurora PostgreSQL PostgreSQL	Sono state aggiunte nuove funzioni al pacchetto di estensione per migliorare la conversione delle tabelle annidate globali. Queste nuove funzioni emulano e DELETE TRIM funzionano EXTEND nel codice sorgente Oracle.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	È stata aggiunta la possibilità di specificare l'ambito di conversione per il codice SQL incorporato nelle applicazioni Java. È ora possibile escludere i sottoinsiemi del progetto dell'applicazione di origine dall'ambito di conversione. Per ulteriori informazioni, consulta Conversione del codice SQL dell'applicazione Java inAWS SCT .
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata degli operatori di concatenazione () all'interno degli indici funzionali.
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle IN condizioni in cui il codice sorgente non include parentesi per una singola espressione.
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle MERGE istruzioni INSERT ON CONFLICT in PostgreSQL.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un errore di analisi che si verificava per i pacchetti di procedure.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale l'azione 5072 appariva inaspettatamente durante la conversione dei pacchetti.
Oracle DW	Amazon Redshift	È stato corretto un errore AWS SCT che impediva l'applicazione del pacchetto di estensione durante l'applicazione del codice convertito al database di destinazione.
Oracle DW	Amazon Redshift	È stato corretto un errore AWS SCT che impediva l'applicazione di alcuni file del pacchetto di estensione durante l'utilizzo della procedura guidata per i pacchetti di estensione.
Oracle DW	Amazon Redshift	È stato risolto un problema a causa del quale AWS SCT non era possibile elaborare la migrazione dei dati AWS Snowball con più di 500 attività in esecuzione in parallelo.
Oracle DW	Amazon Redshift	È stato risolto un problema a causa del quale le funzioni definite dall'utente con tipi definiti dall'utente venivano convertite in modo errato.

Note di rilascio per la build 669 AWS SCT

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	È stato migliorato il processo di valutazione multiserver, che aiuta a determinare la piattaforma di database di destinazione ottimale per i database di origine. Ora, AWS SCT ignora la AWS Secrets Manager chiave se si forniscono le credenziali del database nel file di input con valori separati da virgola (CSV). Per ulteriori informazioni, consulta Creazione di un rapporto di valutazione multiserver .

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	È stato risolto un problema a causa del quale il rapporto di valutazione multiserver includeva l'indirizzo IP del database di origine quando si utilizzava un modulo segreto per la connessione al database AWS Secrets Manager .
Tutti	Amazon Redshift	Implementata la configurazione automatica delle impostazioni della macchina virtuale Java (JVM) in base al sistema operativo e alla RAM disponibile. AWS SCT utilizza questa JVM per eseguire il lavoro degli agenti di estrazione dei dati.
Tutti	Amazon Redshift	Risolto un problema per cui gli agenti di estrazione dei dati non si avviavano in Ubuntu.
Tutti	Amazon Redshift	È stato risolto un problema a causa del quale le attività di estrazione dei dati non si avviavano dopo l'esecuzione del StartAgent .bat file in Windows.
Database SQL di Azure	Aurora PostgreSQL	È stato risolto un problema a causa del quale i nomi delle colonne venivano convertiti in modo errato con l'opzione Genera nomi univoci per gli indici attivata.
Microsoft SQL Server	PostgreSQL	
Greenplum	Amazon Redshift	Conversione implementata delle funzioni che VOID ritornano alle procedure.
Greenplum	Amazon Redshift	È stato risolto un problema a causa del quale la migrazione dei dati non riusciva quando il database di origine includeva valori non numerici (NaN) nelle colonne numeriche. AWS SCT gli agenti di estrazione dati ora sostituiscono i valori NaN con NULL.

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Aggiunta una nuova impostazione di conversione per specificare le TIME FORMAT opzioni DATE FORMAT e durante la conversione delle funzioni integrate. CHAR
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato aggiunto un elemento di azione 8534 per la conversione dei cursori predefiniti dichiarati con la clausola. WITHOUT RETURN Se il cursore non restituisce set di risultati, AWS SCT assegna un NULL valore al nome del cursore nel codice convertito e genera un elemento di azione.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Ha modificato la CURRENT CLIENT_APPLNAME proprietà che identifica AWS SCT durante la connessione al database di origine.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Implementata una nuova impostazione di conversione per specificare le TIME FORMAT opzioni DATE FORMAT e durante la conversione delle funzioni integrate. CHAR
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Conversione implementata delle istruzioni in LEAVE istruzioni di blocco. BEGIN. . .END

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Conversione implementata diXMLPARSE, e funzioniXMLTABLE. XMLNAMESPACES
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle funzioni integrate. CHAR
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata dei cursori.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale l'azione 9996 appariva inaspettatamente durante la conversione delle istruzioni del ciclo. FOR
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata dell'utilizzo dei tipi di tabella nelle istruzioni. SELECT

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	Babelfish per Aurora PostgreSQL	È stato implementato il supporto della nuova versione 2.2.0 del file di configurazione delle funzionalità di Babelfish. Questo file definisce le funzionalità SQL supportate e non supportate da versioni specifiche di Babelfish.
Netezza	Amazon Redshift	Agenti di estrazione dei dati migliorati per risolvere un problema a causa del quale una riga non veniva eliminata dalla tabella di destinazione durante la replica continua dei dati.
Oracle	Amazon RDS per Oracle	Conversione migliorata delle funzionalità di Oracle Database Enterprise Edition.
Oracle	Aurora PostgreSQL PostgreSQL	Conversione delle GROUPING_ID funzioni implementata.
Oracle	Aurora PostgreSQL PostgreSQL	Conversione del codice SQL migliorata nelle applicazioni C# aggiungendo il supporto per la mappatura personalizzata dei tipi di dati nella modalità CLI (Command Line Interface).
Oracle	Aurora PostgreSQL PostgreSQL	È stata migliorata la conversione delle tabelle annidate per evitare un'azione imprevista 9996.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale la chiamata di un costruttore di oggetti veniva convertita in modo errato.
Oracle DW	Amazon Redshift	Supporto implementato delle partizioni di tabella esistenti per la migrazione dei dati. Per accelerare la migrazione dei dati, AWS SCT crea sottoattività per ogni partizione della tabella di origine che non sia vuota. Per ulteriori informazioni, consulta Utilizzo del partizionamento nativo .
Teradata	Amazon Redshift	Conversione migliorata delle CAST funzioni con TIME WITH TIME ZONE AS TIMESTAMP TIME WITH TIME ZONE AS CHAR, e TIMESTAMP AS TIME WITH TIME ZONE argomenti.
Teradata	Amazon Redshift	Conversione migliorata delle CAST funzioni con l'FORMATopzione.
Teradata	Amazon Redshift	È stato risolto un problema a causa CEIL del quale le funzioni non venivano convertite.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale MERGE le istruzioni con DELETE clausole venivano convertite in modo errato.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale TO_CHAR le funzioni con argomenti di data e formato venivano convertite in modo errato.

Note di rilascio per la AWS SCT build 668

Origine	Target	Novità, miglioramento o correzione
Tutti	Amazon Redshift	È stato risolto un problema a causa del quale gli operatori di moltiplicazione nelle regole di migrazione non funzionavano correttamente. Questi operatori consentono di modificare la lunghezza di <code>char</code> ,

Origine	Target	Novità, miglioramento o correzione
		<code>varcharn</code> e <code>varchar</code> , e <code>string</code> i tipi di dati. Per ulteriori informazioni, consulta Creazione di regole di migrazione .
Analisi di Azure Synapse	Amazon Redshift	Supporto implementato di CONVERT funzioni con argomenti. VARCHAR
Analisi di Azure Synapse	Amazon Redshift	Conversione migliorata delle SELECT dichiarazioni con clausole. NOLOCK
Analisi di Azure Synapse	Amazon Redshift	Conversione migliorata delle UPDATE istruzioni con alias o con clausole and. SET FROM
Greenplum	Amazon Redshift	Implementato il partizionamento virtuale automatico per la migrazione e dei dati. AWS SCT utilizza la colonna di GP_SEGMENT_ID sistema per creare partizioni.
Greenplum	Amazon Redshift	Supporto implementato delle RETURN SETOF clausole RETURN QUERY and.
Greenplum	Amazon Redshift	Supporto implementato di SUBSTRING funzioni con tre parametri.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle SUBSTR funzioni con parametri. LOCATE

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Aggiunta un'opzione per utilizzare un array di REFCURSOR variabili per restituire set di risultati dinamici. Quando si seleziona questa opzione nelle impostazioni di conversione, AWS SCT aggiunge un OUT parametro aggiuntivo nel codice convertito.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Supporto implementato delle istruzioni loop. FOR
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Supporto implementato delle funzioni. XMLPARSE È stato aggiunto un elemento di azione 8541 per la separazione degli spazi bianchi nelle XMLPARSE funzioni.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione di più gestori di eccezioni in un unico blocco. BEGIN . . . END
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata di INSERT e trigger. DELETE

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle chiamate di procedura annidata.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata dei tipi di tabella.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale NOT le operazioni logiche bit per bit venivano convertite erroneamente in valori interi.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale gli array locali non venivano inizializzati nella versione 8.0.2 e precedenti di PostgreSQL.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale le MERGE istruzioni con clausole venivano convertite in modo errato. WHEN NOT MATCHED BY SOURCE

Origine	Target	Novità, miglioramento o correzione
MySQL	Aurora MySQL	È stato risolto un problema a causa del quale erano state AWS SCT erroneamente determinate le autorizzazioni utente concesse dal ruolo. <code>rds_superuser_role</code>
Netezza	Amazon Redshift	È stato migliorato il caricatore di metadati di origine per garantire che vengano caricati AWS SCT correttamente gli oggetti del database con nomi in minuscolo.
Oracle	Aurora PostgreSQL L PostgreSQL L	Sono state aggiunte nuove funzioni al pacchetto di estensione per migliorare la conversione delle tabelle annidate locali. Queste nuove funzioni emulano <code>PRIOR</code> , <code>NEXT</code> , <code>LIMIT</code> , <code>FIRST</code> , <code>LAST</code> , <code>EXISTS</code> , <code>EXTEND TRIMDELETE</code> , e <code>SET</code> funzioni presenti nel codice sorgente Oracle. Per ulteriori informazioni, consulta Utilizzo dei pacchetti di estensione .
Oracle	Aurora PostgreSQL L PostgreSQL L	È stata aggiunta la possibilità di specificare l'ambito di conversione per le applicazioni C#. Gli utenti possono ora escludere i sottoinsiemi del progetto dell'applicazione di origine dall'ambito di conversione.
Oracle	Aurora PostgreSQL L PostgreSQL L	Supporto implementato dei <code>COUNT</code> metodi nelle raccolte.
Oracle	Aurora PostgreSQL L PostgreSQL L	Supporto implementato di variabili e costruttori nelle tabelle annidate.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	Supporto implementato di funzioni <code>RATIO_TO_REPORT</code> e funzioni. <code>STANDARD_HASH</code>
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata di oggetti di grandi dimensioni (LOB) come parte del pacchetto di AWS SCT estensione.
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle raccolte locali.
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle JOIN istruzioni con USING clausole in cui i nomi delle colonne non includono il nome della tabella.
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione implementata di funzioni <code>EMPTY_BLOB</code> e <code>EMPTY_CLOB</code>

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione implementata delle variabili di associazione posizionale nelle applicazioni C#.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Conversione implementata di trigger multievento.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Conversione implementata di trigger ricorsivi.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata dei trigger con la variabile globale. @@rowcount
SAP ASE	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale le funzioni aggregate nella SET clausola delle UPDATE istruzioni venivano convertite in modo errato.

Origine	Target	Novità, miglioramento o correzione
SAP ASE	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale l'azione 42702 compariva inaspettatamente durante la conversione delle istruzioni. UPDATE
SAP ASE	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale CONVERT le funzioni con CHAR argomenti venivano convertite in modo errato.
Snowflake	Amazon Redshift	È stato aggiunto il supporto di Snowflake come fonte per la migrazione e dei dati con agenti di estrazione AWS SCT dei dati. Per ulteriori informazioni, consulta Migrazione dei dati da un data warehouse locale ad Amazon Redshift .
Teradata	Amazon Redshift	Conversione migliorata delle CAST funzioni con TIMESTAMP AS TIME WITH TIMEZONE argomenti.

Note di rilascio per AWS SCT Build 667

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	Supporto implementato degli script di estrazione, trasformazione e caricamento (ETL) di Informatica in modalità interfaccia a riga di comando (CLI). AWS SCT reindirizza automaticamente gli script ETL di Informatica al nuovo database di destinazione. Inoltre, AWS SCT converte i nomi degli oggetti e il codice SQL incorporato negli oggetti Informatica. Per ulteriori informazioni, consulta Conversione di script ETL di Informatica .

Origine	Target	Novità, miglioramento o correzione
Tutti	Amazon Redshift	La versione minima del driver supportata per Amazon Redshift è stata aumentata a 2.1.0.9. Per ulteriori informazioni, consulta Scaricamento dei driver di database richiesti .
Azure Synapse Analytics	Amazon Redshift	È stata aggiunta una nuova funzione al pacchetto di estensione per migliorare la conversione della CONVERT funzione con tre argomenti di data e ora.
Analisi di Azure Synapse	Amazon Redshift	Conversione migliorata della funzione. DATEDIFF
Azure Synapse Analytics Microsoft SQL Server DW	Amazon Redshift	Aggiornata la versione del pacchetto di estensione. Assicurati di applicare la versione più recente del pacchetto di estensione nei tuoi AWS SCT progetti esistenti. Per ulteriori informazioni, consulta Utilizzo dei pacchetti di estensione .
BigQuery	Amazon Redshift	È stato risolto un problema a causa del quale gli oggetti filtrati non venivano convertiti nella modalità CLI (Command Line Interface).
Greenplum	Amazon Redshift	È stato corretto un errore a causa del quale AWS SCT non venivano convertite le tabelle temporanee dichiarate in una stored procedure.
Greenplum	Amazon Redshift	È stato corretto un errore a causa del quale gli attributi di codifica delle colonne mancavano nel codice convertito.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	È stata implementata la conversione di UPDATE istruzioni per tabelle autoreferenziali che contengono più di una clausola. INNER JOIN

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Supporto implementato <code>inserted</code> e tabelle <code>deleted</code> temporanee utilizzate da SQL Server per i trigger DML.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Conversione migliorata dei tipi definiti dall'utente nelle stored procedure create in diversi schemi di database. È stato risolto un problema a causa del quale non si AWS SCT riusciva a trovare il tipo di dati e veniva visualizzata l'azione 9996.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	È stato risolto un problema a causa del quale le parentesi quadre (<code>[]</code>) apparivano inaspettatamente attorno ai nomi degli oggetti del database nel codice convertito.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	È stato risolto un problema a causa del quale <code>@@ROWCOUNT</code> le funzioni venivano convertite in modo errato.
Microsoft SQL Server DW	Amazon Redshift	Supporto implementato di <code>geometry</code> e tipi di <code>geography</code> dati.
Microsoft SQL Server DW	Amazon Redshift	Implementato il supporto della <code>MAX</code> parola chiave nelle dichiarazioni dei tipi di dati nel codice convertito.

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server DW	Amazon Redshift	Migliore conversione delle DATEADD funzioni.
Oracle	Aurora PostgreSQL PostgreSQL	Conversione del codice SQL migliorata nelle applicazioni Java aggiungendo il supporto per il MyBatis framework. Per ulteriori informazioni, consulta Conversione del codice SQL nelle applicazioni Java .
Oracle	Aurora PostgreSQL PostgreSQL	Conversione del codice SQL migliorata nelle applicazioni Java che utilizzano il MyBatis framework. È stata aggiunta un'azione 30411 per il codice SQL con sintassi non supportata.
Oracle	Aurora PostgreSQL PostgreSQL	Conversione del codice SQL migliorata nelle applicazioni Pro*C aggiungendo il supporto per le dichiarazioni. typedef struct
Oracle	Aurora PostgreSQL PostgreSQL	Supporto implementato per le dichiarazioni e. CROSS JOIN LEFT JOIN

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	Migliore conversione delle MERGE dichiarazioni. È stato risolto un problema a causa del quale i valori da inserire mancavano nel codice convertito.
Teradata	Amazon Redshift	Sono state modificate le impostazioni di codifica di compressione delle colonne predefinite AWS SCT utilizzate nel codice convertito in modo che corrispondano alle impostazioni predefinite di Amazon Redshift. Per ulteriori informazioni, consulta le codifiche di compressione nella Amazon Redshift Database Developer Guide.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale le operazioni matematiche che utilizzano il tipo di TIME dati venivano convertite in modo errato.
Teradata	RSQL Amazon Redshift	Conversione implementata del FastExport codice contenuto negli script di shell.
Teradata BTEQ	RSQL Amazon Redshift	È stato corretto un errore relativo alla mancata AWS SCT conversione delle istruzioni. COALESCE %data
Vertica	Amazon Redshift	Suggerimenti migliorati per l'ottimizzazione delle conversioni quando un utente seleziona una strategia di ottimizzazione.

Note di rilascio per AWS SCT Build 666

Origine	Target	Novità, miglioramento o correzione
Database SQL di Azure	Aurora PostgreSQL L	È stato risolto un errore di analisi che si verificava per le ON clausole incluse nelle istruzioni. JOIN

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	PostgreSQL	
Analisi di Azure Synapse	Amazon Redshift	Sono state aggiunte tre nuove funzioni al pacchetto di estensione per migliorare la conversione della CONVERT funzione con argomenti di data e ora.
Analisi di Azure Synapse	Amazon Redshift	È stato migliorato il caricatore di metadati di origine per garantire che AWS SCT carichi gli schemi del database di sistema.
Analisi di Azure Synapse	Amazon Redshift	È stato corretto un errore del resolver che si verificava per colonne di tabelle temporanee.
Analisi di Azure Synapse	Amazon Redshift	Conversione implementata dei tipi di VARBINARY dati BINARY e dei tipi di dati nel VARBYTE tipo di dati.
Analisi di Azure Synapse	Amazon Redshift	Supporto implementato del tipo di TIME dati nel codice convertito.
Azure Synapse Analytics	Amazon Redshift	Conversione migliorata delle clausole. COLLATE È stato risolto un problema a causa del quale l'azione 31141 compariva inaspettatamente durante la conversione delle colonne con le regole di confronto predefinite del database.
BigQuery	Amazon Redshift	Conversione implementata delle procedure che modificano i parametri di input.
Greenplum	Amazon Redshift	È stato risolto un problema a causa del quale AWS SCT veniva utilizzata una query non compatibile con i database Greenplum 6.x.

Origine	Target	Novità, miglioramento o correzione
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle EXCEPTION sezioni trasferendo i gestori di eccezioni da Db2 for z/OS a PostgreSQL.
IBM Db2 per z/OS	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle dichiarazioni. OPEN CURSOR
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Conversione implementata delle IIF funzioni utilizzando CASE espressioni.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale le procedure con parametri con valori di tabella venivano convertite in modo errato quando l'CREATE PROCEDURE istruzione non includeva un blocco. BEGIN . . . END
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale la SCOPE_IDENTITY funzione veniva convertita in modo errato.

Origine	Target	Novità, miglioramento o correzione
Oracle	Amazon RDS per Oracle	È stato corretto un errore di caricamento che si verificava con il <code>SELECT_CATALOG_ROLE</code> ruolo durante l'utilizzo di Oracle 10g come sorgente.
Oracle	Amazon RDS per Oracle	È stato migliorato il loader per supportare i job di Oracle Scheduler.
Oracle	Aurora PostgreSQL PostgreSQL	Implementata la conversione di JOIN dichiarazioni con <code>clauseUSING</code> .
Oracle	Aurora PostgreSQL PostgreSQL	Prestazioni migliorate del codice convertito in cui il codice sorgente include variabili globali nelle WHERE clausole.
Oracle	Aurora PostgreSQL PostgreSQL	Conversione del codice SQL migliorata nelle applicazioni Java aggiungendo il supporto per il MyBatis framework. Per ulteriori informazioni, consulta Conversione del codice SQL nelle applicazioni Java .
Oracle DW	Amazon Redshift	Conversione implementata degli operatori PIVOT UNPIVOT relazionali.
Teradata	Amazon Redshift	È stato corretto un errore a causa del quale il codice sorgente che utilizza oggetti JSON non veniva convertito.
Teradata	Amazon Redshift	È stato corretto un errore a causa del quale le tabelle create da un utente eliminato non venivano caricate correttamente.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Conversione implementata delle INSTR funzioni in funzioni native di Amazon RedshiftSTRPOS.
Teradata	Amazon Redshift	Conversione implementata delle TRANSLATE funzioni NVP and.
Teradata	Amazon Redshift	Conversione migliorata delle COALESCE espressioni.
Teradata	Amazon Redshift	Migliore conversione delle DECLARE CONDITION dichiarazioni.
Teradata	Amazon Redshift	Migliore conversione delle EXTRACT funzioni con l'elemento di SECOND sintassi.
Teradata	Amazon Redshift	Migliore conversione delle SQLCODE variabili SQLSTATE and all'interno delle LOOP istruzioni.
Teradata	Amazon Redshift	Conversione migliorata di indici univoci.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale l'azione 9996 compariva inaspettatamente durante la conversione di CURRENT_TIMESTAMP istruzioni con precisione frazionaria impostata su 3.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale le barre rovesciate venivano convertite erroneamente in stringhe letterali.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale EXEC le istruzioni convertite includevano un nome di campo errato nell'istruzione. ADD CONSTRAINT
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale le QUALIFY sottoquery convertite includevano un nome di sottoquery errato.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale le viste convertite non venivano applicate. È stato aggiunto un cast esplicito a un tipo di dati specifico per i valori NULL nel codice convertito.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale le funzioni di data e ora venivano convertite in modo errato.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale le stringhe letterali esadecimali non venivano convertite.

Note di rilascio per la build 665 AWS SCT

Origine	Target	Novità, miglioramento o correzione
Azure Synapse Analytics	Amazon Redshift	Conversione implementata di CONCAT funzioni con argomenti. VARCHAR
Analisi di Azure Synapse	Amazon Redshift	Conversione migliorata delle CREATE TABLE istruzioni che creano tabelle temporanee e non includono il nome dello schema. AWS SCT crea lo dbo schema per memorizzare queste tabelle temporanee nel database di destinazione.
Azure Synapse Analytics	Amazon Redshift	Conversione migliorata delle DROP TABLE istruzioni eseguite su tabelle temporanee.
Analisi di Azure Synapse	Amazon Redshift	Migliore conversione delle OBJECT_ID istruzioni con i blocchi. BEGIN . . . END
Analisi di Azure Synapse	Amazon Redshift	È stato risolto un errore che AWS SCT impediva la conversione delle stored procedure con commenti in blocco.

Origine	Target	Novità, miglioramento o correzione
BigQuery	Amazon Redshift	Conversione implementata dei BigQuery data warehouse in Amazon Redshift. Per ulteriori informazioni, consulta Utilizzo di BigQuery come origine per AWS SCT .
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Conversione migliorata dei trigger che gestiscono più eventi e funzionano con <code>inserted</code> tabelle di <code>deleted</code> sistema in SQL Server.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	È stato corretto un errore del resolver che si verificava per <code>inserted</code> le tabelle di <code>deleted</code> sistema in SQL Server.
Microsoft SQL Server	Babelfish per Aurora PostgreSQL	È stato implementato il supporto della nuova versione 2.1.0 del file di configurazione delle funzionalità di Babelfish. Questo file definisce le funzionalità SQL supportate e non supportate da versioni specifiche di Babelfish.
Oracle	Aurora MySQL MariaDB MySQL	È stato risolto un problema a causa del quale il tipo di <code>varchar2</code> dati veniva convertito in modo errato.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora MySQL	Per i database Oracle versione 12c e successive, AWS SCT supporta i seguenti tipi di dati estesi:
	Aurora PostgreSQL L	<ul style="list-style-type: none"> • VARCHAR2 • NVARCHAR2 • RAW
	MariaDB MySQL PostgreSQL L	AWS SCT ha aumentato la lunghezza massima delle colonne supportata da 8.000 a 32.767 byte per questi tipi di dati.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un errore di analisi che si verificava per il pacchetto Oracle Event Processing.
Teradata	Amazon Redshift	È stata aggiunta l'azione 13214 per più RESET WHEN clause in una singola istruzione. SELECT
Teradata	Amazon Redshift	È stato aggiunto un elemento di azione per SQLSTATE le variabili che si trovano all'esterno di un blocco di gestione delle eccezioni.
Teradata	Amazon Redshift	Conversione implementata delle ACTIVITY_COUNT variabili inROW_COUNT .
Teradata	Amazon Redshift	Conversione implementata della ST_TRANSFORM funzione geometrica incorporata.
Teradata	Amazon Redshift	Conversione migliorata delle istruzioni delete in viste senza la WHERE clausola.
Teradata	Amazon Redshift	Migliore conversione degli CAST operatori nelle espressioni.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Migliore conversione delle GROUP BY clausole.
Teradata	Amazon Redshift	Conversione migliorata INSTR e REGEXP_INSTR funzioni integrate.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale i riferimenti agli alias delle colonne laterali venivano convertiti in modo errato.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale i nomi delle colonne venivano convertiti in modo errato nella sottoquery. QUALIFY
Teradata	Amazon Redshift	Conversione implementata dei .QUIT comandi con la parola chiave ERRORCODE status value.
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto un problema a causa del quale l'azione 9996 compariva inaspettatamente durante la conversione delle dichiarazioni. CREATE
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto un problema a causa del quale l'azione 9998 compariva inaspettatamente durante la conversione delle istruzioni. END

Note di rilascio per la build 664 AWS SCT

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	È stato aggiunto il supporto di Amazon Redshift Serverless come origine e destinazione per progetti di migrazione di database in. AWS SCT Per connetterti ad Amazon Redshift Serverless, assicurati di utilizzare il driver Amazon Redshift JDBC versione 2.1.0.9 o successiva.
Tutti	Tutti	È stata migliorata l'interfaccia utente della finestra delle impostazioni di conversione. AWS SCT ora visualizza le impostazioni solo per le

Origine	Target	Novità, miglioramento o correzione
		coppie di conversione del database con regole di mappatura create. Per ulteriori informazioni, consulta Creazione di regole di mappatura .
Tutti	Tutti	È stato aggiornato il rapporto di valutazione per rimuovere le informazioni duplicate sulla linea e sulla posizione dell'elemento d'azione.
Tutti	Amazon Redshift	Implementato il bilanciamento automatico della memoria nelle attività di estrazione dei dati.
Tutti	Amazon Redshift	È stato risolto un errore a causa del quale gli agenti di estrazione dei dati non potevano connettersi ai AWS Snowball dispositivi.
Database SQL di Azure	Aurora MySQL	Supporto implementato di SUSE Linux 15.3 come piattaforma per eseguire agenti di estrazione dei dati.
IBM Db2 per z/OS	Aurora PostgreSQL	
IBM Db2 LUW	MySQL	
Microsoft SQL Server	PostgreSQL	
MySQL	L	
Oracle		
PostgreSQL		
SAP ASE		

Origine	Target	Novità, miglioramento o correzione
Azure Synapse Analytics	Amazon Redshift	Conversione migliorata delle funzioni. DATEADD
IBM Db2 per z/OS	Aurora PostgreSQL PostgreSQL	È stata aggiunta la possibilità di modificare la classificazione delle colonne nelle regole di migrazione.
Microsoft SSIS	AWS Glue AWS Glue Studio	È stato risolto un errore imprevisto che si verificava quando gli utenti selezionavano uno script sorgente.
Oracle	Aurora MySQL MariaDB MySQL	Conversione implementata dell'utilizzo delle funzioni memorizzate come espressioni di colonna generate. AWS SCT crea trigger per emulare questo comportamento perché MySQL non supporta l'uso di funzioni memorizzate come espressioni di colonna generate.
Oracle	Aurora PostgreSQL PostgreSQL	Conversione implementata delle funzioni dal UTL_MATCH pacchetto come parte del pacchetto di estensione. AWS SCT
Oracle	Aurora PostgreSQL PostgreSQL	Conversione implementata della REGEXP_LIKE funzione con il NULL parametro.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata della SYS_EXTRACT_UTC funzione.
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione del codice SQL migliorata nelle applicazioni C++ implementando il supporto di Wcscats e Wcsncats funzioni. Wcscpys Per ulteriori informazioni, consulta Conversione del codice SQL in applicazioni C++ conAWS SCT .
Oracle DW Snowflake	Amazon Redshift	È stato risolto un problema per cui le istruzioni convertite non includevano la conversione esplicita dei valori nel tipo di dati della colonna. Questo problema si è verificato nelle istruzioni che utilizzano i risultati delle query di altre tabelle.
Teradata	Amazon Redshift	È stata aggiunta la possibilità di modificare la classificazione delle colonne tra case sensitive e case insensitive nelle regole di migrazione. Per ulteriori informazioni, consulta Creazione di regole di migrazione .
Teradata	Amazon Redshift	È stato corretto un errore del resolver che si verificava per le istruzioni i. CREATE TABLE AS
Teradata	Amazon Redshift	È stato corretto un errore a causa del quale la P_INTERSECT funzione incorporata con un'COALESCE espressione non veniva convertita.
Teradata	Amazon Redshift	Conversione implementata delle colonne denominate 0ID_0ID to per evitare l'utilizzo di una parola chiave riservata in Amazon Redshift.
Teradata	Amazon Redshift	Conversione implementata di RENAME istruzioni per funzioni, procedure, viste e macro.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Conversione implementata della STROKE funzione in SPLIT_PART funzione in Amazon Redshift.
Teradata	Amazon Redshift	Conversione migliorata delle funzioni INSTR e REGEXP_INSTR del sistema.
Teradata	Amazon Redshift	Conversione migliorata del tipo di TIME dati.
Teradata	Amazon Redshift	Emulazione migliorata delle MULTISSET tabelle SET e implementando la conversione degli indici univoci primari e secondari.
Teradata	Amazon Redshift	È stato risolto un errore di analisi che si verificava per la funzione. CHARACTER
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto un errore che si verificava quando gli utenti rimuovevano gli script Teradata Basic Teradata Query (BTEQ) dal progetto. AWS SCT

Note AWS SCT di rilascio per Build 663

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	È stata aggiunta la possibilità di modificare la lunghezza di char varcharnvarchar, e string i tipi di dati utilizzando l'operatore di moltiplicazione in una regola di migrazione. Per ulteriori informazioni, consulta Creazione di regole di migrazione .
Tutti	Tutti	Ha implementato il supporto di tre nuove colonne nel rapporto di valutazione multiserver e ha aggiornato il formato del file di input. Assicurati di utilizzare il modello aggiornato del file di input con la versione più recente di. AWS SCT Per ulteriori informazioni, consulta Creazione di un rapporto di valutazione multiserver per la migrazione del database .

Origine	Target	Novità, miglioramento o correzione
Azure Synapse Analytics	Amazon Redshift	Conversione migliorata delle dichiarazioni. OBJECT_ID
Microsoft SQL Server	Babelfish per Aurora PostgreSQL	È stato aggiunto il supporto per Babelfish for Aurora PostgreSQL 1.2.0 come piattaforma di destinazione per i report di valutazione della migrazione dei database. Per ulteriori informazioni, consulta Funzionalità supportate in Babelfish per versione nella Guida per l'utente di Amazon Aurora .
Microsoft SQL Server DW	Amazon Redshift	È stato aggiunto il supporto per le AT TIME ZONE clausole.
Microsoft SQL Server DW	Amazon Redshift	È stato risolto un problema a causa del quale un'istruzione esterna al BEGIN/END blocco veniva convertita in modo errato.
Netezza	Amazon Redshift	Conversione migliorata del tipo di TIME dati e conversione implementata delle relative funzioni, espressioni e valori letterali incorporati.
Oracle	Aurora PostgreSQL PostgreSQL	È stato corretto un errore di caricamento che si verificava quando si utilizzava Oracle 10g come fonte.
Oracle	Aurora PostgreSQL PostgreSQL	Migliore conversione delle clausole OFFSET and. FETCH

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale le procedure con OUT parametri con valori predefiniti venivano convertite in modo errato.
Oracle DW	Amazon Redshift	Conversione migliorata delle funzioni Oracle in funzioni definite dall'utente di Amazon Redshift.
Snowflake	Amazon Redshift	Migliore conversione delle clausole. WITH
Teradata	Amazon Redshift	È stata aggiunta una nuova azione 13209 per i caratteri multibyte non supportati per il tipo di dati. CHAR
Teradata	Amazon Redshift	È stato corretto un errore del caricatore in cui le tabelle non erano completamente caricate.
Teradata	Amazon Redshift	È stato corretto un errore del trasformatore in cui la P_INTERSECT funzione integrata in una JOIN condizione non veniva convertita.
Teradata	Amazon Redshift	È stato risolto un problema per cui il nome di una vista veniva convertito nel caso errato quando l'SELECTistruzione veniva eseguita su una tabella con caratteri speciali nel nome.
Teradata	Amazon Redshift	Conversione migliorata delle INSERT istruzioni con il UNTIL_CHANGED valore nel tipo di PERIOD(DATE) dati.
Teradata	Amazon Redshift	Conversione migliorata della FORMAT funzione integrata utilizzando la TO_CHAR funzione in Amazon Redshift.
Teradata	Amazon Redshift	Conversione migliorata della RANK funzione integrata per garantire che il codice convertito restituisca valori NULL nello stesso ordine del codice sorgente.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Conversione migliorata di vincoli univoci come gli indici univoci primari o secondari.

Note di rilascio per la build 662 AWS SCT

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	È stata aggiunta la possibilità di creare automaticamente AWS SCT progetti per ogni database di origine durante la creazione del rapporto di valutazione multiserver. Con questa opzione attivata, AWS SCT puoi aggiungere regole di mappatura a questi progetti e salvare le statistiche di conversione per l'utilizzo offline. Per ulteriori informazioni, consulta Creazione di un rapporto di valutazione multiserver per la migrazione del database .
Tutti	Tutti	È stato implementato il supporto della percentuale (%) come jolly nei nomi di database e schemi durante la creazione del rapporto di valutazione multiserver.
Tutti	Aurora MySQL Aurora PostgreSQL	Aggiornato il runtime di tutte le AWS Lambda funzioni alla versione 3.9 di Python.
Tutti	Amazon Redshift	Sono stati aggiornati tutti gli agenti di estrazione dati da utilizzare. AWS SDK for Java 2.x
Database SQL di Azure	Aurora PostgreSQL PostgreSQL	Conversione migliorata delle DELETE istruzioni con NON EXISTS clausole.

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server		
Analisi di Azure Synapse	Amazon Redshift	È stato risolto un errore a causa del quale la connessione a un database di origine non riusciva.
IBM Db2 per z/OS	Aurora PostgreSQL PostgreSQL	È stato risolto un errore in cui il codice convertito di un trigger includeva due menzioni dell'alias dell'oggetto.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	È stata migliorata la conversione degli oggetti con nomi in maiuscole e minuscole quando l'opzione Tratta il nome dell'oggetto del database come distinzione tra maiuscole e minuscole è attivata.
Microsoft SQL Server DW Teradata	Amazon Redshift	Conversione implementata degli operatori PIVOT UNPIVOT relazionali.
Netezza	Amazon Redshift	Conversione implementata del tipo di TIME dati.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Implementata la conversione costante UTL_TCP.CRLF del pacchetto.
Oracle	Aurora PostgreSQL PostgreSQL	È stato risolto un problema con il pacchetto di estensione a causa del quale la lunghezza dei tipi di dati per le colonne di lunghezza variabile non veniva mantenuta durante la conversione.
Oracle	Aurora PostgreSQL PostgreSQL	Conversione del codice SQL implementata nelle applicazioni C++. Per ulteriori informazioni, consulta Conversione del codice SQL in applicazioni C++ conAWS SCT .
Oracle	Aurora PostgreSQL PostgreSQL	Supporto implementato della denominazione con distinzione tra maiuscole e minuscole per la conversione di variabili globali e array associativi.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle TO_NUMBER funzioni TO_CHAR TO_DATE, e nel pacchetto di estensione.
Oracle	Aurora PostgreSQL L PostgreSQL L	Migliore conversione dell'TABLE() operatore.
Oracle DW	Amazon Redshift	È stato aggiunto il supporto per la conversione delle chiavi primarie e altri vincoli.
Oracle DW	Amazon Redshift	È stato risolto un problema per cui l'azione 12054 non veniva visualizzata durante la conversione delle istruzioni condizionali.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	È stato risolto un errore che si verificava quando un oggetto con un nome vuoto veniva creato nell'albero di destinazione durante la conversione di tabelle con colonne di tipo definito dall'utente.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	È stato corretto un errore di caricamento per oggetti archiviati come script, routine e così via.
Snowflake	Amazon Redshift	È stato risolto un problema per cui l'azione 22152 non veniva visualizzata quando richiesto e AWS SCT visualizzava il risultato della conversione come commento.

Origine	Target	Novità, miglioramento o correzione
Snowflake	Amazon Redshift	Conversione migliorata delle funzioni di data e ora; supporto implementato dei fusi orari.
Snowflake	Amazon Redshift	È stato risolto un problema a causa del quale le espressioni di tabella comuni (CTE) non ricorsive con una WITH clausola venivano convertite in CTE ricorsive.
Teradata	Amazon Redshift	Conversione migliorata delle istruzioni con collegamenti a tabelle in condizioni. UPDATE
Teradata	Amazon Redshift	Migliore conversione delle RENAME TABLE dichiarazioni.
Teradata	Amazon Redshift	È stato risolto un problema per cui le colonne vuote apparivano nel file con valori separati da virgole (CSV) con un rapporto di valutazione.
Teradata	RSQL Amazon Redshift	È stato corretto un errore a causa del quale mancava un punto e virgola alla fine della macro Basic Teradata Query (BTEQ) convertita.
Teradata	RSQL Amazon Redshift	Conversione migliorata di più valori di tipi di dati nelle istruzioni. CASE
Teradata	RSQL Amazon Redshift	Conversione migliorata della LIKE ANY clausola con un ESCAPE carattere.
Teradata	RSQL Amazon Redshift	Migliore conversione della CAST funzione nelle INSERT istruzioni.
Teradata	RSQL Amazon Redshift	Conversione migliorata dei fusi orari, mappatura delle regioni dei fusi orari implementata.

Origine	Target	Novità, miglioramento o correzione
Teradata	RSQL Amazon Redshift	È stato risolto un problema a causa del quale l'azione 9998 appariva inaspettatamente durante la conversione degli script di shell con script BTEQ.
Teradata	RSQL Amazon Redshift AWS Glue	Implementato il limite di 500 caratteri per i valori delle variabili di sostituzione.
Vertica	Amazon Redshift	Conversione implementata dei tipi di dati BINARY VARBINARY LONG BINARY, BYTEA,, e nel tipo di RAW VARBYTE dati.
Vertica	Amazon Redshift	Conversione migliorata delle funzioni e dei valori letterali incorporati.

Note di rilascio per la AWS SCT build 661

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	Aggiunti filtri per cercare le regole di mappatura nella vista di mappatura. Quando si applica un filtro, AWS SCT visualizza le regole che corrispondono alle condizioni di filtraggio nell'elenco delle mappature del server. Per ulteriori informazioni, consulta Gestione delle regole di mappatura .
Tutti	Tutti	Apache Log4j è stato aggiornato alla versione 2.17.1.
Tutti	Amazon Redshift	È stato aggiunto il supporto per la migrazione dei dati ad Amazon Redshift utilizzando la ENCRYPTED clausola del comando. COPY
Tutti	Amazon Redshift	È stata migliorata l'API REST degli agenti di estrazione dei dati. L'API REST aggiornata aggiunge il supporto di nuove proprietà come la chiave di crittografia, il tipo di crittografia e così via.

Origine	Target	Novità, miglioramento o correzione
Tutti	Amazon Redshift	Implementazione del ruolo negli agenti di estrazione dei dati. Questo aggiornamento migliora la distribuzione delle sottoattività e consente di AWS SCT assegnare attività ai free agent del ruolo specificato.
Tutti	Amazon Redshift	Implementato un controllo che tutti i componenti richiesti siano installati prima dell'applicazione del pacchetto di estensione ad Amazon Redshift.
Analisi di Azure Synapse Microsoft SQL Server DW	Amazon Redshift	Conversione migliorata delle funzioni ERROR_LINE , ERROR_MESSAGE , ERROR_NUMBER , ERROR_PROCEDURE , ERROR_SEVERITY , e di ERROR_STATE sistema per la gestione degli errori.
IBM Db2 per z/OS	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	È stato aggiunto il supporto di IBM Db2 for z/OS versione 12 come fonte per progetti di migrazione di database in. AWS SCT Per ulteriori informazioni, consulta Utilizzo di IBM Db2 for z/OS come fonte .
IBM Db2 LUW	Tutti	È stato migliorato il caricatore di metadati di origine per garantire che AWS SCT carichi i parametri di routine che duplicano i nomi delle colonne.

Origine	Target	Novità, miglioramento o correzione
Database SQL di Microsoft Azure	Aurora PostgreSQL	È stato corretto un errore del trasformatore per le procedure con l'istruzione set. SET NOCOUNT ON
Microsoft SQL Server	PostgreSQL	
Database SQL di Microsoft Azure	Aurora PostgreSQL	Conversione migliorata della CONCAT funzione quando un valore di input è una variabile del tipo definito dall'utente.
Microsoft SQL Server	PostgreSQL	
Database SQL di Microsoft Azure	Aurora PostgreSQL	È stato risolto un problema a causa del quale la DATEPART funzione veniva convertita in modo errato.
Microsoft SQL Server	PostgreSQL	
Microsoft SQL Server	Babelfish per Aurora PostgreSQL	È stato implementato il supporto della nuova versione del file di configurazione delle funzionalità di Babelfish. Questo file definisce le funzionalità SQL supportate e non supportate da versioni specifiche di Babelfish.
Microsoft SQL Server DW	Amazon Redshift	È stato risolto un problema a causa del quale le procedure con un'EXECUTEistruzione venivano convertite in modo errato.

Origine	Target	Novità, miglioramento o correzione
Microsoft SSIS	AWS Glue	È stata migliorata l'interfaccia utente della procedura guidata di configurazione del lavoro. AWS SCT ora visualizza solo le connessioni disponibili nella sezione di configurazione della connessione.
Microsoft SSIS	AWS Glue	È stato risolto un problema a causa del quale le regole di trasformazione non venivano applicate alle attività dei pacchetti e alle regole variabili.
Microsoft SSIS	AWS Glue AWS Glue Studio	È stata aggiunta una nuova azione 25042 per i componenti non supportati.
Microsoft SSIS	AWS Glue Studio	Conversione implementata dei pacchetti di estrazione, trasformazione e caricamento (ETL) di Microsoft SQL Server Integration Services (SSIS) in AWS Glue Studio Per ulteriori informazioni, consulta Conversione di SSIS in AWS Glue Studio .
Oracle	MariaDB	È stato risolto un problema con la conversione dell'OPERATORE MINUS.
Oracle	MariaDB	Conversione migliorata delle funzioni ROWNUM, SYS_GUIDTO_CHAR, e quando la variabile di sql_mode sistema in MariaDB è in Oracle.
Oracle	PostgreSQL	È stata aggiunta un'opzione per evitare la conversione dei tipi di variabili di associazione in tipi SQL in progetti generici di conversione di applicazioni.
Oracle	PostgreSQL	È stata aggiunta un'opzione per evitare di aggiungere il nome dello schema al nome dell'oggetto convertito nei progetti generici di conversione di applicazioni.
Oracle	PostgreSQL	È stato aggiunto il supporto del formato variabile ?x bind per la conversione del codice SQL dell'applicazione.
Oracle DW	Amazon Redshift	Conversione implementata del tipo di RAW dati nel tipo di VARBYTE dati.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Aggiunta un'opzione per emulare le SET tabelle nel codice convertito. Per questa emulazione, AWS SCT supporta MIN e MAX condizioni.
Teradata	Amazon Redshift	Conversione migliorata delle operazioni di join con parametri di diversi tipi di dati. Questo aggiornamento consente di AWS SCT applicare le regole di trasformazione durante la conversione di tali operazioni.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale la GROUP BY clausola veniva convertita in modo errato.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale la QUALIFY clausola veniva convertita in modo errato.
Teradata	Amazon Redshift	È stato risolto un errore imprevisto durante l'importazione FastExport degli script.
Teradata	RSQL Amazon Redshift	Implementata la possibilità di modificare i valori delle variabili in Teradata BTEQ e negli script di shell.
Teradata	RSQL Amazon Redshift	È stato risolto un problema a causa del quale lo script manifest mancava per le sessioni Teradata convertite. FastLoad
Teradata	RSQL Amazon Redshift	È stato risolto un problema a causa del quale l'estensione del file manifest mancava nell'URL (Uniform Resource Locator) per gli script convertiti. FastLoad
Teradata BTEQ	RSQL Amazon Redshift	È stato corretto un errore di caricamento per gli script con variabili di sostituzione.
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto un problema per cui l'azione 27022 non veniva visualizzata quando richiesto.

Note di rilascio per la AWS SCT build 660

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	È stato aggiunto il supporto AWS Secrets Manager e il protocollo Secure Sockets Layer (SSL) nel rapporto di valutazione multiserver. Per ulteriori informazioni, consulta Creazione di un rapporto di valutazione multiserver per la migrazione del database .
Tutti	Tutti	Raccolta di statistiche migliorata per gli oggetti convertiti.
Tutti	PostgreSQL L	Supporto implementato della versione principale 14 di PostgreSQL e MariaDB 10.6 come obiettivi di migrazione.
Analisi di Azure Synapse	Amazon Redshift	Logica di trasformazione migliorata per i nomi degli oggetti convertiti.
Database SQL di Microsoft Azure Microsoft SQL Server	Aurora PostgreSQL L	Conversione migliorata del tipo di XML dati.
Database SQL di Microsoft Azure Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale le NOT LIKE clausole venivano convertite in modo errato.
Database SQL di	Aurora PostgreSQL L	È stato corretto un errore di trasformazione per le procedure con e INSERT le UPDATE istruzioni che DELETE includono la clausola OUTPUT

Origine	Target	Novità, miglioramento o correzione
Microsoft Azure	PostgreSQL L	
Microsoft SQL Server		
Database SQL di Microsoft Azure	Aurora PostgreSQL L	È stato corretto un errore del trasformatore per le procedure con l'istruzione. RETURN @@ROWCOUNT
Microsoft SQL Server	PostgreSQL L	
Microsoft SQL Server	Tutti	Conversione migliorata delle procedure che utilizzano server collegati .
Microsoft SQL Server	Tutti	È stato aggiunto il supporto dell'autenticazione di Microsoft Windows nel rapporto di valutazione multiserver.
Microsoft SQL Server	Aurora PostgreSQL L	È stato corretto un errore di trasformazione per i costruttori di valori di tabella.
	PostgreSQL L	
Microsoft SQL Server DW	Amazon Redshift e AWS Glue	Conversione migliorata degli script di estrazione, trasformazione e caricamento (ETL) per includere il percorso corretto degli script convertiti.

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server DW	Amazon Redshift	È stato risolto un problema a causa del quale venivano generati diversi script convertiti per piattaforme di database di destinazione virtuali e reali.
Oracle	PostgreSQL Aurora PostgreSQL	È stato aggiunto il supporto per la conversione degli indici per le viste materializzate.
Oracle	PostgreSQL Aurora PostgreSQL	È stato risolto un problema a causa del quale l'azione 5982 non veniva visualizzata durante la conversione PRIMARY KEY e UNIQUE limitava l'opzione. NOVALIDATE
Oracle DW	Amazon Redshift	È stato risolto un problema a causa del quale venivano visualizzate categorie aggiuntive nello schema convertito.
Teradata	Amazon Redshift	È stato risolto un problema per cui l'azione 13185 non veniva visualizzata durante la conversione di una colonna non risolta come argomento della funzione. CAST
Teradata	Amazon Redshift	Migliore conversione DELETE e DELETE ALL istruzioni per l'uso del TRUNCATE comando nel codice convertito.
Teradata	Amazon Redshift	Conversione migliorata delle SET tabelle.
Teradata	Amazon Redshift	Migliore conversione delle NORMALIZE condizioni.
Teradata	Amazon Redshift	È stato aggiornato il rapporto di valutazione per rimuovere le statistiche di conversione dello schema del database dall'elenco degli oggetti di archiviazione del database.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Conversione migliorata della UPDATE dichiarazione senza la FROM clausola.
Teradata	Amazon Redshift	Implementato il supporto del tipo di VARBYTE dati nel codice convertito.
Teradata BTEQ	AWS Glue	Risolto un problema per cui l' AWS Glue opzione Converti in era disattivata nel menu contestuale.
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto un problema a causa del quale i tipi di dati mancavano nel codice convertito.
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto un problema a causa del quale le variabili di sostituzione venivano citate erroneamente nel codice convertito.
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto un problema con la conversione delle variabili di sostituzione con valori negli script. FastLoad
Vertica	Amazon Redshift	Implementato il supporto del tipo di TIME dati nel codice convertito.
Vertica	Amazon Redshift	Conversione migliorata delle ORDER BY espressioni SELECT DISTINCT e delle espressioni.
Vertica	Amazon Redshift	È stato aggiunto il supporto per la conversione dei vincoli.
Vertica	Amazon Redshift	È stato risolto un errore a causa del quale un rapporto di valutazione non veniva salvato come file con valori separati da virgole (CSV).

Note di rilascio per la build 659 AWS SCT

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	È stata migliorata la procedura guidata per il nuovo progetto che genera un rapporto di valutazione combinato per più database di origine.
Tutti	Tutti	È stato risolto un problema per cui il pacchetto di estensione non veniva creato in progetti che includevano più database di origine e di destinazione.
Tutti	Tutti	Conversione migliorata del codice SQL incorporato nel codice sorgente dell'applicazione.
Tutti	Tutti	È stata aggiunta la possibilità di eseguire script da diverse cartelle nell'interfaccia della AWS SCT riga di comando.
Tutti	Amazon Redshift	È stato migliorato il messaggio di avviso fornito quando gli utenti scelgono Esegui l'ottimizzazione nei progetti di migrazione con la piattaforma di database di destinazione virtuale Amazon Redshift.
Tutti	Aurora PostgreSQL	Supporto implementato della versione principale 13 di PostgreSQL su Aurora PostgreSQL Compatible Edition come obiettivo di migrazione.
Tutti	Amazon RDS per MySQL	Per impostazione predefinita, è stata implementata la conversione del codice senza distinzione tra maiuscole e min
Analisi di Azure Synapse	Amazon Redshift	È stato risolto un errore a causa del quale la connessione a un database di origine non riusciva nell'interfaccia della riga di comando.
Microsoft SQL Server	PostgreSQL	Conversione migliorata delle procedure che includono UPDATE istruzioni con condizioni di unione.

Origine	Target	Novità, miglioramento o correzione
	Aurora PostgreSQL L	
Microsoft SQL Server	PostgreSQL L Aurora PostgreSQL L	Conversione migliorata dei trigger, delle stored procedure e delle funzioni che includono il valore dopo il segno di uguale.
Microsoft SQL Server	PostgreSQL L Aurora PostgreSQL L	È stato corretto un errore del trasformatore per le procedure con l'DELETEistruzione e l'ORoperatore.
Microsoft SQL Server	PostgreSQL L Aurora PostgreSQL L	Migliore conversione della OUTPUT clausola.
Microsoft SQL Server DW	Amazon Redshift e AWS Glue	Conversione migliorata del tipo di NUMERIC dati.
Microsoft SQL Server DW	Amazon Redshift	Conversione migliorata delle viste che hanno un alias di tabella con lo stesso nome della tabella originale.
Microsoft SSIS	AWS Glue	È stato risolto un problema per cui le credenziali di AWS Glue connessione non venivano visualizzate nella finestra Configura connessioni.

Origine	Target	Novità, miglioramento o correzione
Netezza	Amazon Redshift	È stata aggiunta la possibilità di ripetere ogni giorno l'esecuzione delle attività di migrazione dei dati di modifica dei dati di acquisizione dei dati (CDC).
Netezza	Amazon Redshift	È stato risolto un problema per cui la scheda Attività diventava inattiva dopo l'annullamento della registrazione di un agente di estrazione dati.
Netezza	Amazon Redshift	È stato risolto un problema per cui la conferma della registrazione dell'agente di migrazione dei dati non veniva visualizzata nell'interfaccia utente.
Netezza	Amazon Redshift	È stato risolto un problema a causa del quale una connessione a un database di origine non riusciva con un errore Loader.
Netezza	Amazon Redshift	È stato risolto un errore a causa del quale gli agenti di migrazione dei dati non venivano eseguiti dopo l'apertura di un progetto salvato.
Oracle	Amazon RDS per Oracle	Supporto implementato di Oracle Unified Auditing.
Oracle	PostgreSQL L Aurora PostgreSQL L	Conversione del codice SQL implementata nelle applicazioni C#. Per ulteriori informazioni, consulta Conversione del codice SQL in applicazioni C# .
Oracle	PostgreSQL L Aurora PostgreSQL L	Implementata una nuova logica di trasformazione per i nomi degli oggetti con distinzione tra maiuscole e minuscole per migliorare la visibilità delle modifiche alla conversione del codice. AWS SCT converte i nomi degli oggetti in maiuscolo in minuscolo. È vero anche il contrario; AWS SCT converte i nomi degli oggetti da lettere minuscole a maiuscole. I nomi di altri oggetti e le parole riservate vengono convertiti senza modifiche.

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL Aurora PostgreSQL	Conversione migliorata delle partizioni hash senza il NOT NULL vincolo.
Oracle	Aurora PostgreSQL	È stato aggiunto il supporto per la conversione di Oracle e CHECK i NOT NULL vincoli FOREIGN KEY relativi alla clausola. ENABLE NOVALIDATE
Oracle DW	Amazon Redshift	È stato risolto un problema a causa del quale venivano migrati i valori errati per i numeri in virgola mobile.
Oracle DW	Amazon Redshift e AWS Glue	È stato risolto un problema relativo alle colonne vuote nel rapporto di valutazione della migrazione del database in un file con valori separati da virgole (CSV).
SAP ASE	PostgreSQL Aurora PostgreSQL	È stato risolto un problema relativo a un'interruzione imprevista della conversione.
Snowflake	Amazon Redshift	Conversione migliorata del tipo di VARIANT dati.
Teradata	Amazon Redshift	Migliore conversione della COLLECT STATISTICS dichiarazione.
Teradata	Amazon Redshift	È stato risolto un problema per cui l'azione 9998 non veniva visualizzata durante la conversione di viste nidificate con colonne. PERIOD
Teradata	Amazon Redshift e AWS Glue	È stato risolto un problema per cui una piattaforma di AWS Glue destinazione virtuale non veniva visualizzata nell'interfaccia utente dopo l'apertura di un progetto salvato.

Origine	Target	Novità, miglioramento o correzione
Teradata BTEQ	AWS Glue	È stato risolto un problema per cui la conversione in una piattaforma di AWS Glue destinazione virtuale non era supportata dopo l'apertura di un progetto salvato.
Teradata BTEQ	RSQL Amazon Redshift	Evidenziazione della sintassi migliorata del codice convertito.
Teradata BTEQ	RSQL Amazon Redshift	Implementato il controllo dei valori dei parametri dopo il caricamento. I valori non supportati sono evidenziati nella scheda Variabili.
Vertica	Amazon Redshift	Conversione implementata delle funzioni aggregate.
Vertica	Amazon Redshift	Implementata la conversione delle proiezioni in viste materializzate e migliorata l'interfaccia utente che mostra il codice sorgente delle proiezioni.

Note di rilascio per la build 658 AWS SCT

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	Integrazione fornita con AWS Secrets Manager. È ora possibile utilizzare le credenziali di connessione al database archiviate in Secrets Manager.
Tutti	Tutti	È stato aggiunto il supporto per gli script in formato YAML nell'interfaccia a riga di comando. AWS SCT
Tutti	Amazon Redshift	Supporto implementato degli endpoint di interfaccia Amazon S3 (VPCE) negli agenti di estrazione dei dati.

Origine	Target	Novità, miglioramento o correzione
Tutti	Amazon Redshift	È stato aggiunto il supporto per la piattaforma di database di destinazione virtuale Amazon Redshift oltre alla combinazione e alla combinazione di Amazon AWS Glue Redshift già supportate.
Greenplum	Amazon Redshift	È stato risolto un problema a causa del quale l'opzione Salva come SQL non salvava il codice SQL convertito in un file.
IBM Db2 LUW Database SQL di Microsoft Azure Microsoft SQL Server Oracle SAP ASE	Aurora MySQL	Conversione migliorata per supportare nuove funzionalità di Amazon Aurora MySQL Compatible Edition con compatibilità MySQL 8.0.
Microsoft SQL Server	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	È stato risolto un problema per cui l'azione 810 non veniva visualizzata quando richiesto.

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Migliore conversione delle procedure con UPDATE e DELETE delle INSERT istruzioni.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema per cui l'azione 7810 non veniva visualizzata quando richiesto.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata di un'EXECistruzione annidata all'interno di un'IF . . . ELSE istruzione.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle viste indicizzate.
Netezza	Amazon Redshift	Agenti di migrazione dei dati migliorati grazie al monitoraggio delle transazioni in tempo reale a pieno carico nell'operazione Change Data Capture (CDC). Ora puoi interrompere le attività di migrazione dei dati se la sessione CDC è pianificata per iniziare in un determinato momento. Inoltre, puoi visualizzare il livello di registrazione degli errori nella console dopo aver interrotto un'operazione con CDC.
Oracle	Tutti	È stato migliorato il caricatore di tabelle per garantire che AWS SCT carichi oggetti con opzioni di condivisione.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	È stata migliorata la conversione della SYSDATE funzione e aggiunta la possibilità di modificare il fuso orario nelle impostazioni di conversione.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale le istruzioni dinamiche non venivano convertite.
Oracle	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema per cui il codice convertito non includeva i nomi generati dal sistema.
Oracle Oracle DW	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle SELECT istruzioni annidate all'interno dei trigger.
Oracle DW	Amazon Redshift	Conversione migliorata delle TO_DATE TO_TIMESTAMP_TZ funzioni e del pacchetto di estensione. TO_TIMESTAMP
Snowflake	Amazon Redshift	È stata aggiunta un'opzione per salvare il codice SQL convertito in file diversi per ogni oggetto o per ogni istruzione.
Teradata	Amazon Redshift	Conversione migliorata della CONCAT funzione.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Conversione migliorata di un'SELECTistruzione annidata all'interno di una WHERE clausola.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale SET le MULTISSET tabelle e venivano convertite in modo errato dopo che gli utenti eliminavano e ricreavano una tabella.
Teradata	Amazon Redshift	Migliore conversione delle procedure che includono una WITH clausola.
Teradata	Amazon Redshift	Conversione migliorata del tipo di DATE dati.
Teradata	RSQL Amazon Redshift	È stato risolto un problema a causa del quale si verificava un errore imprevisto del trasformatore durante la conversione FastExport degli script.
Teradata BTEQ	RSQL Amazon Redshift	È stato aggiunto il supporto per la conversione di un indice di unione in una vista materializzata.
Teradata BTEQ	RSQL Amazon Redshift	È stato aggiunto il supporto per la conversione di una TITLE definizione che include più righe.
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto un problema a causa del quale la dimensione di un tipo di dati geospaziali non veniva convertita.
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto un problema a causa del quale i nomi dei parametri venivano convertiti in caratteri minuscoli.
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto un problema che impediva la conversione di una stored procedure annidata all'interno di un'MACROistruzione.

Origine	Target	Novità, miglioramento o correzione
Vertica	Amazon Redshift	Migliore conversione dell'ALLoperatore.
Vertica	Amazon Redshift	È stato risolto un problema per cui l'Use Union all view?opzione nelle impostazioni di conversione non veniva applicata.
Vertica	Amazon Redshift	Conversione migliorata dei tipi di TIME WITH TIMEZONE dati TIME e.
Vertica	Amazon Redshift	È stato risolto un problema con il caricamento delle tabelle flessibili.

Problemi risolti:

- Miglioramenti generali.

Note di rilascio per la AWS SCT build 657

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	Apache Log4j è stato aggiornato alla versione 2.17 per risolvere problemi di vulnerabilità di sicurezza.
Tutti	Amazon Redshift	Progetti di ottimizzazione dello schema migliorati, in cui le statistiche di gestione delle chiavi non venivano salvate nel progetto. AWS SCT
Amazon Redshift	Amazon Redshift	È stato risolto un problema con l'aggiornamento delle informazioni sul server.
Apache Cassandra	Amazon DynamoDB	È stato risolto un problema con le regole di mappatura quando si utilizzava l'interfaccia a riga di comando AWS SCT .
Apache Cassandra	Amazon DynamoDB	È stato risolto un problema che impediva la creazione dell'attività di migrazione a causa di un titolo aggiornato nel certificato.

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	È stato risolto un problema che impediva la visualizzazione dell'azione 7672 durante la conversione delle procedure di Microsoft SQL Server con SQL dinamico.
Database SQL di Azure	Aurora PostgreSQL	Conversione migliorata delle funzioni con valori di tabella.
Microsoft SQL Server	PostgreSQL	
Database SQL di Azure	Aurora PostgreSQL	È stato risolto un problema a causa del quale l'OUTargomento in una stored procedure con il valore restituito predefinito non veniva convertito nell'INOUTargomento.
Microsoft SQL Server	PostgreSQL	
Greenplum	Amazon Redshift	Strategie di ottimizzazione migliorate trovando nella QueryLog tabella le tabelle e le colonne più utilizzate.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Sono stati risolti i problemi relativi alla conversione di quanto segue: <ul style="list-style-type: none"> Operatore di assegnazione di concatenazioni di stringhe () += Funzione SCOPE_IDENTITY Tipo di dati varchar(max)

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle visualizzazioni con funzioni non supportate.
Microsoft SQL Server	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema a causa del quale le funzioni non supportate come argomento di un'altra funzione venivano convertite in modo errato.
Microsoft SQL Server	Babelfish per Aurora PostgreSQL L	Conversione migliorata dei riferimenti alle tabelle di transizione.
Microsoft SQL Server DW	Amazon Redshift	È stata aggiunta la categoria delle funzioni di aggregazione all'albero dei metadati del database di origine.
Microsoft SQL Server DW	Amazon Redshift	Conversione migliorata del tipo di TIME dati.

Origine	Target	Novità, miglioramento o correzione
Azure Synapse Analytics Greenplum Netezza Microsoft SQL Server DW Snowflake Teradata	Amazon Redshift	È stato risolto un problema a causa del quale DROP CREATE gli script non venivano salvati quando si utilizzava una piattaforma di database di destinazione virtuale.
Servizi di integrazione Microsoft SQL Server	AWS Glue	È stato risolto un problema a causa del quale gli script degli oggetti di origine non venivano visualizzati nell'interfaccia utente.
Netezza	Amazon Redshift	Strategie di ottimizzazione migliorate scegliendo la tabella dei fatti e le dimensioni appropriate per la collocazione.
Oracle	Aurora PostgreSQL PostgreSQL	È stato risolto un problema relativo alla corretta conversione dei trigger Oracle, che utilizzano numeri di sequenza.

Origine	Target	Novità, miglioramento o correzione
Oracle	Aurora PostgreSQL L PostgreSQL L	Conversione migliorata delle visualizzazioni con collegamenti a database pubblici.
Oracle DW	Amazon Redshift	Strategie di ottimizzazione migliorate analizzando la cardinalità delle colonne dell'indice.
Oracle DW	Amazon Redshift	È stato risolto un problema a causa del quale le funzioni scalari personalizzate definite dall'utente con concatenazione di stringhe venivano convertite in modo errato.
Snowflake	Amazon Redshift	È stato risolto un problema per cui l'opzione Salva come SQL non veniva visualizzata nell'interfaccia utente.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale la raccolta delle statistiche non riusciva con l'LOADER ERROR eccezione.
Teradata	Amazon Redshift	È stato risolto un problema per cui l'opzione Crea rapporto non veniva visualizzata nell'interfaccia utente.
Teradata	Amazon Redshift	Conversione migliorata della CAST funzione.
Teradata	Amazon Redshift	Risolto un problema di conversione per ST_Line_Interpolate_Point .
Teradata	Amazon Redshift	È stato rimosso un valore imprevisto dal percorso della libreria Python.
Teradata	RSQL Amazon Redshift	È stato corretto un errore del resolver che si verificava durante la conversione di più script. FastLoad

Origine	Target	Novità, miglioramento o correzione
Teradata BTEQ	RSQL Amazon Redshift	Conversione migliorata dei tipi di dati di DATABASE comando e geometria.
Teradata BTEQ	AWS Glue	È stato risolto un problema di sincronizzazione errata degli script di origine e destinazione nell'interfaccia utente.

Problemi risolti:

- Miglioramenti generali.

Note di rilascio per la AWS SCT build 656

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	È stato aggiunto il supporto di più database di origine e destinazione all'interno di un unico progetto. Gli utenti possono ora creare regole di mappatura per abbinare diversi schemi di database e piattaforme di destinazione nello stesso progetto.
Tutti	Tutti	È stato aggiunto il supporto di piattaforme di database di destinazione virtuali. Gli utenti ora non hanno bisogno di connettersi a un database di destinazione per vedere come AWS SCT converte lo schema del database di origine.
Tutti	Tutti	<p>Miglioramenti dell'interfaccia utente:</p> <ul style="list-style-type: none"> • Aggiunte le opzioni Connect to the server e Disconnect from the server agli alberi di metadati di origine e destinazione. • È stata aggiunta un'opzione per rimuovere un server di database dal AWS SCT progetto.

Origine	Target	Novità, miglioramento o correzione
Cassandra	Amazon DynamoDB	È stato risolto un problema di ricerca in cui la CASSANDRA_HOME variabile non includeva una barra (/) dopo <code>cassandra.yaml</code> la conf cartella.
Cassandra	Amazon DynamoDB	È stato aggiunto il supporto di Amazon Machine Image (AMI) per Amazon Linux 2.
Cassandra	Amazon DynamoDB	Messaggio di errore migliorato fornito quando viene fornita una chiave errata per Cassandra.
Cassandra	Amazon DynamoDB	Conversione migliorata modificando una proprietà nel <code>cassandra-env.yaml</code> file in base alla versione del database di destinazione.
Cassandra	Amazon DynamoDB	La versione Java sulla destinazione Cassandra Datacenter è stata aumentata alla 1.8.0.
Greenplum	Amazon Redshift	Strategie di ottimizzazione migliorate nelle impostazioni del progetto.
Greenplum	Amazon Redshift	È stato risolto un problema di migrazione dei dati in cui gli oggetti non venivano applicati al database con questo errore: <code>An I/O error occurred while sending to the backend</code> .
Greenplum Microsoft SQL Server DW	Amazon Redshift	È stato risolto un problema per cui l'Apply <code>RTRIM to string columns</code> opzione non veniva visualizzata nell'interfaccia utente.
Microsoft SQL Server	Babelfish per Aurora PostgreSQL	È stato aggiunto il supporto per Babelfish for Aurora PostgreSQL come piattaforma di destinazione. Gli utenti possono ora creare un rapporto di valutazione per stimare la migrazione da SQL Server a Babelfish per Aurora PostgreSQL.
Netezza	Amazon Redshift	Strategie di ottimizzazione migliorate nelle impostazioni del progetto.

Origine	Target	Novità, miglioramento o correzione
SAP ASE	Aurora PostgreSQL L PostgreSQL L	Implementata la capacità di generare nomi univoci per gli indici.
SAP ASE	Aurora PostgreSQL L PostgreSQL L	È stato risolto un problema con una colonna di indice duplicata nello script di destinazione.
Snowflake	Amazon Redshift	È stato risolto un problema a causa del quale le opzioni Nascondi schemi vuoti, Nascondi database vuoti e Nascondi database/schemi di sistema non venivano visualizzate nell'interfaccia utente.
Teradata	RSQL Amazon Redshift	È stato aggiunto il supporto per la conversione degli script di MultiLoad lavoro Teradata in script RSQL di Amazon Redshift.
Teradata	RSQL Amazon Redshift	È stato risolto un problema con la conversione delle variabili di sostituzione negli script and. FastLoad FastExport
Teradata	RSQL Amazon Redshift	È stato risolto un problema per cui le azioni non venivano visualizzate nella scheda Azioni dopo il passaggio dalla scheda Riepilogo.
Teradata	RSQL Amazon Redshift	È stato risolto un problema a causa del quale si verificava un errore dopo la generazione del rapporto durante la conversione FastExport degli script.
Teradata	RSQL Amazon Redshift	Sono stati risolti i problemi di formattazione dopo la conversione degli script di shell.

Origine	Target	Novità, miglioramento o correzione
Teradata	RSQL Amazon Redshift	È stato risolto un problema per cui AI 13177 ora viene commentato nello script convertito.
Teradata	Amazon Redshift	Risolto un problema di conversione delle tabelle temporali.
Teradata	Amazon Redshift	Migliore conversione della SET QUERY_BAND dichiarazione.
Teradata	Amazon Redshift	Risolto un problema di conversione dell'NORMALIZE operazione.
Vertica	Amazon Redshift	Migliorata la descrizione di AI 17008.

Problemi risolti:

- Miglioramenti generali.

Note di rilascio per la AWS SCT build 655

Origine	Target	Novità, miglioramento o correzione
Teradata	RSQL Amazon Redshift	È stato risolto un problema per garantire che tutti i problemi di valutazione compaiano nei report quando FastLoad o MultiLoad vengono utilizzati.
Teradata	RSQL Amazon Redshift	È stato aggiunto il supporto per la conversione degli script di FastExport lavoro Teradata in script RSQL di Amazon Redshift.
Teradata	RSQL Amazon Redshift	È stato risolto un problema per garantire che l'azione Salva manifesto in S3 fosse abilitata in modalità offline durante l'utilizzo. FastLoad

Origine	Target	Novità, miglioramento o correzione
Teradata	RSQL Amazon Redshift	È stato risolto un problema per garantire l'applicazione delle regole di mappatura per script come FastLoad
Greenplum	Amazon Redshift	La versione minima del driver supportata per Greenplum è stata aumentata a 42.2.5.
Greenplum	Amazon Redshift	È stata aggiunta una connessione a Greenplum tramite SSL con la versione del driver 42.2.5 o successiva.
Oracle DW	Amazon Redshift	Supporto migliorato per l'esecuzione di funzioni scalari personalizzate definite dall'utente (UDF) all'interno di un'altra UDF.
Oracle DW	Amazon Redshift	È stato risolto un problema per cui le funzioni non venivano applicate al database con questo errore: Failed to compile udf
Oracle DW	Amazon Redshift	Conversione migliorata utilizzando dichiarazioni di tipo appropriate, ad esempio pls-type per i parametri %ROWTYPE.
Teradata	RSQL Amazon Redshift	È stato risolto un problema a causa del quale i problemi di valutazione del tipo di informazione non venivano visualizzati nel rapporto.
Teradata	RSQL Amazon Redshift	Risolto un errore del trasformatore dopo la conversione di alcuni script.
Teradata	RSQL Amazon Redshift	È stato risolto un problema che ora viene commentato nello script convertito.
Teradata	Amazon Redshift	Risolto un problema per cui FastExport ->EXPORT -> 'null' veniva visualizzato al posto di 'CAST' dopo la conversione.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Risolto un problema per cui alcune funzioni di un pacchetto di estensione non funzionavano quando veniva applicata con la versione del driver Cause: <code>[JDBC Driver]String index out of range: 0 1.2.43</code>
Teradata	Amazon Redshift	SET table conversion: aggiunta l'emulazione della tabella SET per le istruzioni insert-select.
Teradata	Amazon Redshift	CAST: supporta il casting di tipi di dati aggiuntivi.
Teradata	Amazon Redshift	Risolto un problema di conversione per «other_current_time_01»
Teradata	Amazon Redshift	Teradata FastExport — Amazon Redshift RSQL: conversione migliorata dei comandi Teradata—Field FastExport
Teradata	Amazon Redshift	Teradata FastExport — Amazon Redshift RSQL: conversione migliorata dei comandi Teradata—Layout FastExport
Oracle	PostgreSQL Aurora PostgreSQL	È stato risolto un problema a causa del quale lo script di destinazione degli oggetti con SAVE EXCEPTIONS STATEMENT veniva modificato dopo la riconversione.
Oracle	PostgreSQL Aurora PostgreSQL	È stato risolto un problema a causa del quale, dopo la conversione, veniva specificato un campo errato nella clausola ORDER BY. <code>proc_cursor_with_calc_columns</code>

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL Aurora PostgreSQL	Risolto: è richiesta una dichiarazione di <code>aws_oracle_ext\$array_id\$temporary</code> variabile aggiuntiva in una conversione ASSOCIATIVE COLLECTION.
Oracle	PostgreSQL Aurora PostgreSQL	Risolto: conversione errata di una CHIAVE PRIMARIA con lo stesso nome di un INDICE di proprietà della stessa tabella.

Problemi risolti:

- Miglioramenti generali.

Note di rilascio per la AWS SCT build 654

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL Aurora PostgreSQL	È stato risolto un problema con Hierarchical Query Pseudocolumns, errore di analisi delle colonne PRIOR.
Oracle	PostgreSQL Aurora PostgreSQL	È stato risolto un problema relativo alla conversione corretta di un commento su più righe contenente una barra e un asterisco (/ *).

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL L Aurora PostgreSQL L	È stata aggiunta l'emulazione USER_COL_COMMENTS per la visualizzazione di sistema al pacchetto di estensione.
Oracle	PostgreSQL L Aurora PostgreSQL L	Conversione migliorata dei valori letterali tra virgolette.
DB2 LUW	PostgreSQL L Aurora PostgreSQL L	Conversione migliorata delle istruzioni LABEL che aggiungono o sostituiscono etichette nelle descrizioni di tabelle, viste, alias o colonne.
Oracle	Nessuno	Tabella di sistema SYS.USER\$ sostituita con visualizzazione DBA_USERS e query migliorate.
Oracle DW	Amazon Redshift	Interrogazioni sui metadati Oracle DW aggiornate.
Teradata	RSQL Amazon Redshift	È stato aggiunto il supporto per la conversione di script shell, Teradata FastLoad e Teradata Basic Teradata Query (BTEQ) in script RSQL di Amazon Redshift.
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto il problema per cui «merge_01" veniva convertito in modo errato.

Origine	Target	Novità, miglioramento o correzione
Teradata BTEQ	RSQL Amazon Redshift	È stato risolto il problema che faceva apparire End or Identify (EOI) alla fine di uno script su una nuova riga.
Azure Synapse	Amazon Redshift	Messaggio di errore migliorato fornito quando viene fornita una password errata per Azure Synapse.
Teradata	Amazon Redshift	È stata migliorata la conversione delle istruzioni UPDATE per mantenere l'alias corretto in base allo standard Teradata.
Teradata	Amazon Redshift	È stato risolto un errore di conversione del cursore a causa del quale le azioni non venivano ricevute.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale una conversione TD_NORMALIZE_OVERLAP eliminava le righe.
Teradata	Amazon Redshift	Ora supporta il controllo rigoroso della data per la funzione TO_DATE migliorata.
Teradata	Amazon Redshift	Conversione migliorata della funzione integrata TO_NUMBER (n).
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale la categoria Schemas era assente dall'albero dei metadati.
Greenplum	Amazon Redshift	Aggiunta la selezione GP_SEGMENT_ID all'elenco durante la creazione di una partizione virtuale per una tabella Greenplum.
Greenplum	Amazon Redshift	Risolto un problema a causa del quale le funzioni non venivano applicate sulla destinazione.
MS SQL Server DW	Amazon Redshift	Risolto un problema a causa del quale si verificava un errore di trasformazione dopo la conversione senza AI 9996.
MS SQL Server DW	Amazon Redshift	È stato risolto un problema a causa del quale veniva registrato un errore durante l'apertura della procedura guidata del pacchetto di estensione.

Origine	Target	Novità, miglioramento o correzione
MS SQL Server DW	Amazon Redshift	È stato risolto un problema che si verificava quando veniva utilizzato uno stile di commenti errato per le funzioni Redshift Python.
Netezza	Amazon Redshift	È stato risolto un problema che impediva la creazione di un pacchetto di estensione Netezza-Redshift con un AWS profilo.
Teradata	RSQL Amazon Redshift	Conversione migliorata del comando SESSIONS. FastLoad
Teradata	RSQL Amazon Redshift	Rapporti di valutazione FastLoad degli script migliorati.
Teradata	RSQL Amazon Redshift	Azione FastLoad WRITER Save to S3 implementata.
Teradata	RSQL Amazon Redshift	Risolto un problema per cui FastLoad i pulsanti Save Script\ Save manifest to s3 non erano attivi.
Teradata	RSQL Amazon Redshift	È stato risolto un problema per cui FastLoad multifile_script creava un solo file manifest dopo la conversione anziché i tre file previsti.
Teradata	RSQL Amazon Redshift	È stato risolto un problema a causa della visualizzazione di cartelle aggiuntive in un percorso S3. FastLoad
Teradata	RSQL Amazon Redshift	È stato risolto un problema a causa del quale il nome del file manifesto FastLoad era errato in un percorso S3.

Problemi risolti:

- Miglioramenti generali.

Note di rilascio per la AWS SCT build 653

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL L Aurora PostgreSQL L	Implementata la capacità di convertire SQL dinamico creato in funzioni o procedure chiamate.
Oracle	PostgreSQL L Aurora PostgreSQL L	Conversione SQL dinamica migliorata: parametri IN-Parametri come variabili di associazione.
Oracle DW 18, 19	Amazon Redshift	Miglioramenti alla conversione da Oracle a Redshift implementati: conversioni integrate avanzate. LISTAGG aggregato; LISTAGG analitico.
Oracle DW 18,19	Amazon Redshift	Miglioramenti alla conversione da Oracle a Redshift implementati: interroga nuove funzionalità.
Vertica	Amazon Redshift	Miglioramenti alla conversione da Vertica a Redshift implementati: connessione da SSL a JDBC con SSL=true.
MS SQL Server DW	Amazon Redshift	Miglioramenti alla conversione da MS SQL Server a Redshift: tabelle esterne.
Teradata	Amazon Redshift	Miglioramenti alla conversione da Teradata a Redshift: tipi di dati INTERVAL, operazioni aritmetiche.
Teradata	Amazon Redshift	Miglioramenti alla conversione da Teradata a Redshift: supporto per alias a colonna laterale.
Oracle	Nessuno	Le seguenti query Loader ora utilizzano al posto di: DBA_USERS SYS.USER\$

Origine	Target	Novità, miglioramento o correzione
		<ul style="list-style-type: none"> • <code>get-tree-path-list- .sql by-name-path</code> • <code>estimate-table-or-view- .sql constraints-by-schema</code> • <code>estimate-table-or-view- .sql constraints-by-selected-schemas</code>
Teradata	Amazon Redshift	Migliore allineamento dei commenti quando SCT converte le macro Teradata in stored procedure Redshift.
Oracle DW	Amazon Redshift	Conversione migliorata degli elementi di formato Date/Timestamp: <code>TO_DATE</code> , <code>TO_TIMESTAMP</code> , <code>TO_TIMESTAMP_TZ</code>
Teradata	Amazon Redshift	Errore di conversione del cursore Teradata risolto.
Teradata	Amazon Redshift	È stato risolto il problema che causava l'eliminazione degli attributi <code>TD_NORMALIZE_OVERLAP</code> durante la conversione.
Teradata	Amazon Redshift	Risolto un problema per cui <code>MAX</code> la funzione veniva ignorata quando SCT convertiva una query.
Teradata	Amazon Redshift	SCT ora converte la funzione Teradata <code>CHARACTERS</code> nella funzione Redshift <code>LENGTH</code> .
Teradata	Amazon Redshift	SCT ora supporta la conversione di <code>FORMAT</code> in <code>TO_CHAR</code> per i formati più comunemente usati.
Tutti	Tutti	Conversione migliorata delle routine crittografate.

Problemi risolti:

- Miglioramenti generali.

Note di rilascio per la AWS SCT build 652

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	PostgreSQL	Aggiunto il blocco <code>sp_getapplock</code> e le <code>sp_releaseapplock</code> funzioni dell'app.
Nessuno	Amazon Redshift	Miglioramento dell'interfaccia CLI (Command Line Interface): implementata la modalità Script Command.
Oracle	PostgreSQL Aurora PostgreSQL	Implementato il campionamento dei parametri di routine all'interno di SQL dinamico.
Oracle	PostgreSQL Aurora PostgreSQL	Miglioramenti alla conversione in SQL dinamico creato in funzioni o procedure chiamate.
Microsoft SQL Server Oracle DB2 LUW	Aurora PostgreSQL	Ogni funzione lambda viene implementata e configurata tramite policy una sola volta e le funzioni lambda comuni vengono riutilizzate per tutte le fonti possibili.
DB2 LUW	PostgreSQL	È stato risolto il problema che causava il messaggio di errore «9996 — Severity critical — Si è verificato un errore del trasformatore» quando si utilizzava DB2 LUW come sorgente.
Teradata	Amazon Redshift	Support per le espressioni di tabella ricorsive nel prossimo lancio di Amazon Redshift.

Origine	Target	Novità, miglioramento o correzione
Azure Synapse	Amazon Redshift	Regole di ottimizzazione dello schema implementate.
Teradata	Amazon Redshift	Supporta la conversione del fuso orario dalle macro Teradata alle stored procedure Redshift.
Teradata	Amazon Redshift	Supporta l'aritmetica sui valori PERIOD.
Teradata	Amazon Redshift	Supporta la conversione delle espressioni di tabella comuni ricorsive di Teradata (RECURSIVE CTE).
Teradata	Amazon Redshift	Supporta gli identificatori con distinzione tra maiuscole e minuscole tramite le impostazioni utente, <code>enable_case_sensitive_identifier</code> . Quindi, «COLUMN_NAME» e «Column_Name» diventano nomi di colonna diversi.
Teradata	Amazon Redshift	È stato risolto il problema relativo al tipo di dati decimale in modo che i campi decimali vengano convertiti con la stessa precisione.
Teradata	Amazon Redshift	È stato risolto il problema relativo alla conversione aritmetica degli intervalli in modo che la sottrazione aritmetica degli intervalli venga convertita correttamente.
Teradata	Amazon Redshift	Cast di tipo Teradata NUMBER to DATE migliorato.
Teradata	Amazon Redshift	Casting di tipo Teradata DATE to NUMBER migliorato
Teradata BTEQ	Amazon Redshift	Conversione dei tipi di dati PERIOD migliorata.
Teradata	Amazon Redshift	È stato risolto il problema relativo al caricamento dei metadati per una tabella con colonne GEOMETRY, che ora viene caricata correttamente da Teradata.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Supporta la conversione delle istruzioni di fusione durante la conversione delle macro Teradata in stored procedure Redshift.
Teradata	Amazon Redshift	Conversione migliorata di semplici macro durante la migrazione da Teradata a Redshift.
Teradata	Amazon Redshift	Garantito che la conversione delle istruzioni Teradata UPDATE riporti il nome alias corretto in base allo standard Teradata.

Problemi risolti:

- Miglioramenti generali.

Note di rilascio per la AWS SCT build 651

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	AWS SCT Rapporti migliorati per aggiornare i link alle azioni di conversione consigliate elencate.
MS SQL Server	PostgreSQL	Aggiunto il supporto per la conversione della STR() funzione.
MS SQL Server	PostgreSQL	È stato aggiunto il supporto per la conversione dell'operatore EXOR bit per bit (in ^ Microsoft SQL Server) in PostgreSQL come operatore . #
Oracle	PostgreSQL	Risolto un problema per cui la <code>aws_oracle_ext.UNI STR(null)</code> funzione AWS SCT del pacchetto di estensione si bloccava NULL su una destinazione PostgreSQL. AWS SCT ora gestisce il NULL
Teradata BTEQ	RSQL Amazon Redshift	Miglioramenti alla conversione apportati per risolvere un problema a causa del quale la conversione di Amazon Redshift RSQL MERGE generava un errore di trasformazione.

Origine	Target	Novità, miglioramento o correzione
Oracle DW	Amazon Redshift	Integrati migliorati implementati.
Oracle DW	Amazon Redshift	Sono stati aggiunti miglioramenti basati sulle funzionalità di metadati, tra cui il partizionamento Auto-List (TBL_PART_LIST_AUTO), l'elenco a più colonne (TBL_PART_MULTI_LIST) e il riferimento a intervalli (TBL_PART_RANGE_INTVAL_REF).
nessuno	Amazon Redshift	Aumento dei limiti della tabella delle partizioni UNION ALL delle partizioni fisiche utilizzate per le conversioni.
Teradata	Amazon Redshift	Miglioramenti delle conversioni apportati all'ambito dei rapporti di valutazione.
Teradata	Amazon Redshift	Miglioramenti apportati alle conversioni MACRO complesse di Teradata.
Teradata	Amazon Redshift	Conversione migliorata delle macro Teradata in procedure archiviate di Amazon Redshift con commenti su SQL non supportati.
Teradata	Amazon Redshift	È stato risolto un problema a causa del quale la conversione delle macro Teradata in procedure memorizzate di Amazon Redshift generava riferimenti errati ai nomi alias.
Teradata	Amazon Redshift	Conversione migliorata della dichiarazione Teradata. QUALIFY
Teradata	Amazon Redshift	Conversione migliorata per trasferire i commenti ad Amazon Redshift e conservare una cronologia delle modifiche apportate alla vista.
Teradata	Amazon Redshift	È stato risolto un problema per cui la clausola RESET WHEN non produceva la conversione corretta.
Teradata BTEQ	Amazon Redshift	Conversione migliorata degli script BTEQ che contengono istruzioni MERGE.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Sono state aggiunte funzioni integrate per migliorare la conversione dei campi del tipo di dati PERIOD.
Microsoft SQL Server	Amazon Redshift	Mappatura migliorata del tipo di dati di trasformazione per il tipo di dati TIME.
Tutti	Tutti	Aggiunto l'accesso alla pubblicazione iniziale del manuale di riferimento AWS Schema Conversion Tool CLI in formato PDF. Vedi AWS Schema Conversion Tool CLI Reference .

Problemi risolti:

- Miglioramenti generali.

Note di rilascio per AWS SCT Build 650

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	<p>Uso aggiornato e migliorato degli agenti estrattori, tra cui:</p> <ul style="list-style-type: none"> • Una configurazione da utilizzare con l'archiviazione condivisa e un agente di copia dedicato. • Esportazione e importazione di attività di estrazione dei dati da un progetto all'altro. • Support per Azure SQL Data Warehouse (Azure Synapse) come sorgente. • Utilizzo del partizionamento Netezza nativo.

Origine	Target	Novità, miglioramento o correzione
		Per ulteriori informazioni, consulta Migrazione dei dati da un data warehouse locale ad Amazon Redshift .
Tutti	Amazon RDS PostgreSQL 13	AWS SCT ora supporta Amazon RDS PostgreSQL 13 come destinazione.
Microsoft SQL Server	Aurora PostgreSQL	Conversione migliorata di un set di risultati da una procedura Microsoft SQL Server a un target Aurora PostgreSQL.
Oracle DW	Amazon Redshift	Sono stati implementati miglioramenti alla conversione da Oracle ad Amazon Redshift.
Oracle DW	Amazon Redshift	Miglioramenti implementati alla conversione di istruzioni SQL dinamiche.
Oracle DW	Amazon Redshift	Miglioramenti implementati alla conversione SQL UDF.
Oracle DW	Amazon Redshift	Messaggio chiarito che AWS SCT non supporta la conversione di TABELLE ESTERNE.
Oracle DW	Amazon Redshift	Funzioni di conversione integrate avanzate.
Teradata BTEQ	RSQL Amazon Redshift	Migliore gestione dei parametri di sostituzione all'interno degli script BTEQ durante l'utilizzo della GUI. AWS SCT

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server DW	Tutti	È stata aggiornata la versione minima del driver JDBC supportata per Microsoft SQL Server, Azure, Azure Synapse.
Microsoft SQL Server		
Azzurro		
Azure Synapse		

Problemi risolti:

- Teradata: miglioramenti aggiuntivi alla conversione delle macro [RISOLTO]
- I caratteri speciali sono sfuggiti alla destinazione causando errori SQL ed è stata necessaria una nuova elaborazione per reinserirli [RISOLTO]
- Miglioramenti generali

Note di rilascio per la AWS SCT build 649

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server DW	Amazon Redshift	Miglioramenti alla conversione da MSSQL ad Amazon Redshift per supportare le tabelle temporali.
Oracle DW	Amazon Redshift	Miglioramenti delle funzioni integrati implementati, come: <ul style="list-style-type: none"> Funzioni di conversione <ul style="list-style-type: none"> • TO_BINARY_DOUBLE •

Origine	Target	Novità, miglioramento o correzione
		<p>TO_BINARY_FLOAT</p> <ul style="list-style-type: none"> • TO_NUMBER • TO_DATE • TO_TIMESTAMP • TO_TIMESTAMP_TZ • TO_DSINTERVAL • TO_YMINTERVAL • VALIDATE_CONVERSION
Oracle DW	Amazon Redshift	<p>Miglioramenti delle funzioni implementati per l'elaborazione approssimativa delle interrogazioni, come:</p> <p>Funzioni di aggregazione</p> <ul style="list-style-type: none"> • ANY_VALUE • APPROX_COUNT_DISTINCT • APPROX_COUNT_DISTINCT_DETAIL • APPROX_COUNT_DISTINCT_AGG • LISTAGG • TO_APPROX_COUNT_DISTINCT

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Miglioramenti di conversione implementati per l'ordinamento automatico e la selezione delle chiavi di distribuzione di Teradata. Il motore DB seleziona automaticamente le chiavi di distribuzione e ordinamento. È stato introdotto un pulsante di opzione denominato Usa la sintonizzazione automatica della tabella di Amazon Redshift sulle impostazioni dei progetti correnti > Strategie di ottimizzazione > Strategie di selezione delle chiavi iniziali.
Teradata	Amazon Redshift	Caricatore di AWS SCT tabelle migliorato per garantire il caricamento di tutte le tabelle da AWS SCT Teradata.
Teradata	Amazon Redshift	Sono stati implementati miglioramenti alla conversione in modo che Amazon Redshift supporti modelli di subquery correlati che includono una semplice clausola WHERE NOT EXISTS.
Teradata	Amazon Redshift	È stato aggiunto il supporto per l'uso dei comandi ECHO nelle macro.
DB2 LUW	PostgreSQL L Aurora PostgreSQL L	Supporto implementato per la conversione DYNAMIC RESULTS SETS, tra cui: <ul style="list-style-type: none"> • Clausola cursore WITH RETURN/WITH RETURN TO CLIENT • DYNAMIC RESULT SETS (conversione delle claus
Microsoft SQL Server	Aurora PostgreSQL L	Supporto implementato per l'attuale Aurora RDS PostgreSQL come destinazione.
Oracle		
DB2 LUW		
SAP ASE		

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server Oracle DB2 LUW SAP ASE	MariaDB	Supporto implementato per Mariadb 10.5 come destinazione.
Microsoft SQL Server	MariaDB	Implementato il supporto di INSERT-RETURNING che restituisce un set di risultati delle righe inserite.
Oracle	Aurora PostgreSQL	È stato aggiunto il supporto della funzione XMLFOREST per la conversione da Oracle a Aurora PostgreSQL.

Problemi risolti:

- Miglioramenti generali.

Note di rilascio per la AWS SCT build 648

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL Amazon Aurora PostgreSQL-Compatible Edition	Implementata la modalità di applicazione personalizzata del pacchetto di estensione Aurora PostgreSQL: operatori per tipi numerici/data e testo.

Origine	Target	Novità, miglioramento o correzione
Oracle Microsoft SQL Server DB2 LUW	Aurora PostgreSQL	<p>Configurazione Aurora PostgreSQL Lambda Invoke implementata: creazione dell'estensione <code>aws_lambda</code>; assegnazione del ruolo IAM al cluster Aurora PostgreSQL.</p> <ul style="list-style-type: none"> • Oracle WebAgent: e-mail, lavori, code, file • DB2: e-mail, attività, file • Microsoft SQL Server: e-mail, agente
Oracle	PostgreSQL	<p>Il refactoring della conversione delle istruzioni FORALL è stato implementato:</p> <ul style="list-style-type: none"> • Dichiarazione FORALL • PER TUTTI... SALVA ECCEZIONI • RITORNO CON BULK COLLECT • raccolta di sistema <code>SQL%BULK_EXCEPTIONS</code>
Oracle DW18, 19	Amazon Redshift	Miglioramenti alla conversione da Oracle ad Amazon Redshift implementati: conversioni integrate avanzate. LISTAGG aggregato; LISTAGG analitico.
Oracle DW 18,19	Amazon Redshift	Miglioramenti alla conversione da Oracle ad Amazon Redshift implementati: interroga nuove funzionalità.
Vertica	Amazon Redshift	Miglioramenti alla conversione da Vertica ad Amazon Redshift implementati: connessione da SSL a JDBC con <code>SSL=true</code> .

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server DW	Amazon Redshift	Miglioramenti alla conversione da Microsoft SQL Server a Redshift: tabelle esterne.
Teradata	Amazon Redshift	Miglioramenti alla conversione da Teradata a Redshift: tipi di dati INTERVAL, operazioni aritmetiche.
Teradata	Amazon Redshift	Miglioramenti alla conversione da Teradata a Redshift: supporto per alias a colonna laterale.

Problemi risolti:

- Miglioramenti generali

Note di rilascio per la AWS SCT build 647

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	Microsoft SQL Server	RDS ora supporta la funzionalità Database Mail.
Microsoft SQL Server	MySQL	<p>Implementazione del nome massimo di ogni tipo di identificatore: la lunghezza massima dei nomi degli oggetti (ad esempio tabelle, vincoli, colonne) in SQL Server è di 128 caratteri. La lunghezza massima dei nomi degli oggetti in MySQL è di 64 caratteri. Per scrivere oggetti convertiti nel database MySQL è necessario abbreviarne i nomi. Per evitare la duplicazione dei nomi dopo il taglio, è necessario aggiungere il «checksum» del nome dell'oggetto originale ai nuovi nomi.</p> <p>Taglia i nomi più lunghi di 64 caratteri come segue:</p> <pre>[first N chars]() + "" + [checksum]()</pre>

Origine	Target	Novità, miglioramento o correzione
		<p>[first N chars] = 64 - 1 - [length of checksum string]</p> <p>Per esempio: example_of_a_test_schema_with_a_name_length_g reater_than_64_characters ?? example_of_a_test _schema_with_a_name_length_greater_than_64_97 03</p>
Oracle	MySQL/ Aurora MySQL	Implementati il caricamento e la conversione dei commenti sugli oggetti di archiviazione. Ad esempio, l'elaborazione dei commenti sulle tabelle e l'elaborazione dei commenti sulle colonne Table/View.
Teradata	Amazon Redshift	È stato aggiunto il supporto per la conversione del tipo di dati TIME.
Teradata	Amazon Redshift	Miglioramenti alla conversione: implementato TD_NORMALIZE_OVERLAP.
Microsoft SQL Server DW	Amazon Redshift	Miglioramenti alla conversione: SELECT con clausola WITH; SELECT senza FROM
Tutti	Tutti	AWS SCT Data Migration Service Assessor (DMSA): questa nuova funzionalità consente di valutare più server e ricevere un rapporto di riepilogo che mostra la migliore direzione di destinazione per l'ambiente.
Tutti	Tutti	AWS SCT Wizard: il confronto degli obiettivi ora mostra le differenze tra gli obiettivi in un'unica visualizzazione tabellare.
Tutti	Tutti	Interfaccia utente Tree Filter: il filtro dei metadati riprogettato gestisce modelli di filtraggio più complessi.
Tutti	Tutti	Rapporto di valutazione: la sezione Avviso riprogettata fornisce una descrizione migliore e una comprensione più chiara di un problema.

Problemi risolti:

- Miglioramenti generali
- Estrattori di dati: operazione secondaria non riuscita con `ConcurrentModificationException` [RISOLTO].
- Da Microsoft SQL Server a MySQL: lunghezze massime degli identificatori [RISOLTO].

Note di rilascio per Build 646 AWS SCT

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL	Implementazione migliorata del modello in formato TM.
Oracle	PostgreSQL	L'implementazione della maschera in formato SP fornisce il supporto di base per il suffisso SP, solo per la lingua inglese.
Oracle	PostgreSQL	Gestione dei nomi di oggetti lunghi di Oracle: AWS SCT ora gestisce i nomi di oggetti lunghi Oracle in base all'attributo <code>Target Max Identifier Length</code> .
	Amazon Redshift	Codifica Amazon Redshift AZ64 con AWS SCT — Aggiunta la codifica di compressione AZ64 per alcuni tipi di dati
Teradata	Amazon Redshift	È stato aggiunto il supporto per la conversione delle transazioni implicite.
Teradata	Amazon Redshift	È stato aggiunto il supporto per le funzioni geospaziali integrate di Teradata: Metodi <code>ST_LineString</code>
Greenplum	Amazon Redshift	Conversione della sequenza Greenplum: aggiunti gli elementi successivi alle schede Proprietà: valore minimo, valore massimo, incremento, ciclo.
Greenplum	Amazon Redshift	Resolver: aggiunta la risoluzione del tipo di dati «char».

Origine	Target	Novità, miglioramento o correzione
Greenplum	Amazon Redshift	Lunghezza di conversione dei caratteri: conversione PL/pgSQL aggiornata per il tipo di carattere.
Greenplum	Amazon Redshift	Risolto un problema con la selezione della chiave di distribuzione Greenplum a causa del quale una tabella aveva la CHIAVE DI DISTRIBUZIONE ma non AWS SCT riusciva a riconoscere e recuperare la tabella come DISTRIBUITA CASUALMENTE.
Teradata	Amazon Redshift	Supporto per cursori Teradata — Aggiunto il supporto per la conversione dei cursori.
Teradata	Amazon Redshift	Colonne Identity: è stato aggiunto il supporto per la conversione delle colonne Identity.
Teradata	Amazon Redshift	Tipi di dati INTERVAL: è stato aggiunto il supporto per la conversione dei tipi di dati INTERVAL.

Problemi risolti:

- Miglioramenti generali
- Greenplum: impossibile eseguire la conversione a causa dell'errore nel registro [RISOLTO].
- MSSQL — PostgreSQL: errore del trasformatore durante la conversione della funzione LAG [RISOLTO].
- MSSQL — PostgreSQL: SCOPE_IDENTITY [RISOLTO].
- AWS SCT sospensione nei progetti DW [RISOLTO].
- È necessaria una regola di mappatura per rimuovere spazio aggiuntivo sul nome della colonna in AWS SCT [RISOLTO].

Note di rilascio per AWS SCT Build 645

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Fornisci una soluzione per risolvere le viste Teradata non completamente qualificate (nomi di visualizzazione o oggetti non completamente qualificati all'interno della vista).
Teradata	Amazon Redshift	Aggiunto il supporto della funzione ASCII per i nodi di calcolo.
Teradata	Amazon Redshift	Quando AWS SCT individua dati multibyte in un Teradata CHAR definito come CHAR(N), questi vengono convertiti in Amazon VARCHAR(3*N) Redshift.
Teradata	Amazon Redshift	Fornisci la conversione CAST Teradata tra date e numeri. <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	Supporta la conversione dei tipi di PERIOD dati Teradata in due colonne Amazon TIMESTAMP Redshift: <ul style="list-style-type: none">• <code>PERIOD(TIMESTAMP)</code>• <code>PERIOD(TIMESTAMP WITH TIMEZONE)</code>
Teradata	Amazon Redshift	Supporta la conversione della RANK funzione Teradata con RESET WHEN clausola.
Teradata	Amazon Redshift	Supporto migliorato di CAST nelle conversioni esplicite di tipi di dati e CAST impliciti nelle espressioni.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	Segnala modelli di sottoquery correlati non supportati. Per ulteriori informazioni, consulta la sezione Subquery correlate nella Amazon Redshift Database Developer Guide.
nessuno	Amazon Redshift	Le tabelle migliorate limitano il supporto per i tipi di nodi RA3.
Teradata	Amazon Redshift	È stato aggiunto il supporto per l'estrazione di dati geospaziali Teradata. Per ulteriori informazioni, consulta la sezione Interrogazione di dati spaziali in Amazon Redshift nella Amazon Redshift Database Developer Guide .
Microsoft SQL Server	PostgreSQL	È stata aggiunta l'opzione, <code>. convert_procedures_to_function</code>

Problemi risolti:

- Miglioramenti generali

Note di rilascio per la AWS SCT build 644

Le modifiche per le AWS SCT versioni 1.0.643 vengono unite nella versione 1.0.644. AWS SCT

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	<p>Molteplici miglioramenti alla conversione.</p> <ul style="list-style-type: none"> • Conversioni migliorate QUALIFY con l'alias della tabella. • Conversioni migliorate con l'operatore. IN • Conversione migliorata con l'LIKEoperatore. •

Origine	Target	Novità, miglioramento o correzione
		<p>Conversioni migliorate con evidenziazione dei problemi nel codice convertito.</p> <ul style="list-style-type: none">• Conversioni migliorate con QUALIFY clausole di ordine insolito di WHERE, in SQL.• Sono stati corretti gli errori del trasformatore durante la conversione della procedura di JOIN() costruzione. UPD_FT_SV C_TRANS_BH_CBH_IND• Migliore conversione delle macro in procedure memorizzate. <p>Sono stati aggiunti comandi AWS SCT CLI speciali in grado di analizzare gli script sql/bteq forniti e generare un rapporto sul numero di strutture sintattiche rilevate nel codice sorgente.</p> <ul style="list-style-type: none">• Numero di comandi BTEQ• Numero di GESTORI• Numero di casi CAST• Numero di casi DML/DDD• Numero di DML nelle visualizzazioni aggiornabili <p>È stato aggiunto un elemento d'azione per il report di valutazione: le colonne Teradata con formati di data personalizzati non sono supportate in Amazon Redshift.</p>

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Aggiunta funzionalità per salvare gli script di installazione dei pacchetti di estensione.</p> <p>Livello di gravità modificato per AI 5334.</p> <p>Prestazioni migliorate grazie all'utilizzo di un record come variabile IMPLEMENTATION di pacchetto.</p> <p>Aggiunto il supporto per le funzioni XMLAGG di aggregazione</p>
IBM Db2	PostgreSQL/Aurora PostgreSQL	<p>Aggiunto il caricamento e la conversione dei commenti sull'implementazione degli oggetti di archiviazione.</p>
MS SQL DW	Amazon Redshift	<p>Miglioramento della conversione: problema risolto con PATINDEX.</p> <p>Miglioramenti dell'interfaccia utente:</p> <ul style="list-style-type: none"> • Salva come SQL per l'implementazione dell'albero dei sorgenti. • È stata aggiunta una logica aggiuntiva alla generazione di script per più file.
Vertica	Amazon Redshift	<p>Miglioramento dell'interfaccia utente: Salva come SQL per l'implementazione dell'albero dei sorgenti.</p>

Problemi risolti:

- Miglioramenti generali alle conversioni tra Teradata e Amazon Redshift
- Correzione di bug generali e miglioramenti dell'interfaccia utente

Note di rilascio per la AWS SCT build 642

Modifiche per la AWS Schema Conversion Tool versione 1.0.642.

Note

AWS Schema Conversion Tool (AWS SCT) le modifiche alla build 1.0.642 sono applicabili a Windows, Ubuntu e Fedora. Non esiste una build 1.0.642 per macOS.

Origine	Target	Novità, miglioramento o correzione
Microsoft SSIS	AWS Glue	Conversione implementata dei pacchetti ETL di Microsoft SQL Server Integration Services (SSIS) in AWS Glue. Per ulteriori informazioni, consulta Conversione di SSIS in AWS Glue con AWS SCT .
Oracle	Modalità MariaDB/SQL = Oracle/MySQL/Amazon Aurora MySQL	Implementata la sezione di dichiarazione PL/SQL nella clausola WITH.
Oracle	PostgreSQL/Aurora PostgreSQL	Aggiunta del supporto per <code>DBMS_SESSION.RESET_PACKAGE</code> e <code>DBMS_SESSION.MODIFY_PACKAGE</code> .
Vertica	Amazon Redshift	Abilita l'esportazione di script SQL da un database Vertica ad Amazon Redshift.

Problemi risolti:

- Miglioramento del rapporto di valutazione.
- Miglioramento dell'interfaccia utente del rapporto di valutazione.
- Aggiungo la possibilità di modificare le impostazioni JVM dall'interfaccia utente.
- Miglioramenti generali.

Note di rilascio per la AWS SCT build 641

Modifiche per la AWS Schema Conversion Tool versione 1.0.641.

Note

AWS Schema Conversion Tool (AWS SCT) le modifiche alla build 1.0.641 sono applicabili a Windows, Ubuntu e Fedora. Non esiste una build 1.0.641 per macOS.

Origine	Target	Novità, miglioramento o correzione
Oracle/ MS SQL/ MySQL/ PostgreSQL/ DB2 LUW	Tutti	Produci i calcoli del Time Report nel file.csv.
Teradata	Amazon Redshift	È stato aggiunto il supporto per la funzione CSUM. È stato aggiunto il supporto per i tipi di dati geospaziali Teradata.
Teradata	Tutti	È stato aggiunto il supporto per la conversione delle colonne IDENTITY.
Greenplum	Amazon Redshift	Aggiunto il supporto per lo stile di distribuzione AUTO durante la conversione della tabella Greenplum.
SAP ASE	Tutti	Produci i calcoli del Time Report nel file.csv.

Risoluzioni:

- Risoluzione di vari bug.
- Vari miglioramenti delle prestazioni.

Note di rilascio per AWS SCT Build 640

Le modifiche per le AWS SCT versioni 1.0.633, 1.0.634, 1.0.635, 1.0.636, 1.0.637, 1.0.638, 1.0.639 e 1.0.640 vengono unite nella versione 1.0.640. AWS SCT

Note

AWS SCT le modifiche alla build 1.0.640 sono applicabili a Windows, Ubuntu e Fedora. Non si applicano a macOS.

Non puoi installare la AWS SCT versione 1.0.640 o successiva su Apple macOS. AWS SCT la versione 1.0.632 è stata l'ultima versione a supportare l'installazione su Apple macOS.

Nelle tabelle seguenti, è possibile trovare elenchi delle funzionalità e correzioni di bug per le versioni AWS Schema Conversion Tool che sono state combinate nella release 1.0.640. Queste tabelle raggruppano caratteristiche e correzioni di bug dal motore di origine.

Argomenti

- [Modifiche alla release 1.0.640 di Oracle](#)
- [Modifiche alla release 1.0.640 di Microsoft SQL Server](#)
- [Cambiamenti di MySQL versione 1.0.640](#)
- [Modifiche a PostgreSQL nella release 1.0.640](#)
- [Modifiche alla versione 1.0.640 Db2 LUW](#)
- [Modifiche alla versione 1.0.640 di Teradata](#)
- [Modifiche alla versione 1.0.640 per altri motori](#)

Modifiche alla release 1.0.640 di Oracle

Nella tabella seguente sono elencate le modifiche di generazione 1.0.640 in cui Oracle è il motore di origine.

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL L	Conversione del codice SQL implementata nelle applicazioni Java e Pro*C.

Origine	Target	Novità, miglioramento o correzione
	Aurora PostgreSQL L	
Oracle	PostgreSQL L Aurora PostgreSQL L	<p>Miglioramento delle prestazioni delle seguenti funzioni quando utilizzate in una clausola 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	Aggiunto il supporto RDS MariaDB 10.4 per tutti i fornitori di elaborazione transazionale online (OLTP).
Oracle	PostgreSQL L/Aurora PostgreSQL L	<p>Aggiunto il supporto per <code>DBMS_UTILITITIT.GET_TIME</code>.</p> <p>Aggiunte le seguenti emulazioni:</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>

Origine	Target	Novità, miglioramento o correzione
Oracle	MariaDB/MySQL/Aurora MySQL/modalità Microsoft SQL Server=Oracle/PostgreSQL/Aurora PostgreSQL/RDS Oracle	Aggiunto supporto clausola di condivisione per TABLE(DATA, EXTENDED DATA), VIEW(DATA, EXTENDED DATA) e SEQUENCE(DATA)
Oracle	PostgreSQL/Aurora PostgreSQL/Oracle RDS	<p>La definizione DEFAULT di una colonna può essere estesa in modo che DEFAULT venga applicato per l'inserimento NULL esplicito.</p> <p>La clausola DEFAULT ha una nuova clausola ON NULL. Questa nuova clausola indica al database di assegnare un valore di colonna predefinito specificato quando un'istruzione INSERT tenta di assegnare un valore che restituisce NULL.</p>
Oracle	MariaDB/MariaDB (SQL MODE=ORACLE)	Aggiunto il supporto per "Colonne identità", che incrementano automaticamente al momento dell'inserimento.
Tutti	Tutti	Aggiornamento ad Amazon Corretto JDK 11 da JDK 8. Per ulteriori informazioni, inclusi i link per il download, consulta Cos'è Amazon Corretto 11? nella Guida per l'utente di Amazon Corretto 11.
Tutti	Tutti	Sono state aggiunte informazioni al report di valutazione sulle possibili incongruenze nel database dell'utente.

Origine	Target	Novità, miglioramento o correzione
Oracle	MariaDB 10.2/MariaDB 10.3/ MySQL/ Aurora MySQL/ PostgreSQL/ Aurora PostgreSQL	La clausola DEFAULT dispone di una nuova clausola ON NULL, che indica al database di assegnare un valore di colonna predefinito specificato quando un'istruzione INSERT tenta di assegnare un valore che restituisce NULL.
Oracle	Oracle RDS/ MySQL /Aurora MySQL/ PostgreSQL/ Aurora PostgreSQL	Aggiunto supporto per le colonne IDENTITY.
Oracle	MySQL 8.x	Aggiunto il supporto per il vincolo CHECK.

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Implementato il controllo di ANYDATA IS NULL/IS NOT NULL utilizzando la routine degli extension pack.</p> <p>Implementata l'emulazione della funzione VALUE utilizzata in una query basata sulla funzione TABLE di XMLSequence.</p> <p>Aggiunto il supporto DBMS_LOB per le seguenti routine predefinite:</p> <ul style="list-style-type: none"> • DBMS_LOB.CREATETEMPORARY • DBMS_LOB.FREETEMPORARY • DBMS_LOB.APPEND
Tutti	SQL Server	<p>SQL Server 2019: aggiunto il supporto per il nuovo attributo indice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: aggiunto il supporto per i tipi di tabella Nodo e Edge Database Graph.</p> <p>SQL Server 2016: aggiunto il supporto per TABLES TEMPORAL.</p>
Tutti	Tutti	<p>Implementata la possibilità di sovrascrivere le partizioni fisiche con partizioni virtuali. Gli estrattori di data warehouse estraggono i dati in base alle partizioni virtuali create.</p>
Oracle	Amazon Redshift	<p>Implementata la conversione degli attributi del cursore nei blocchi nidificati.</p> <p>Amazon Redshift non supporta le raccolte. Le variabili correlate vengono convertite come VARCHAR. Tutte le operazioni di raccolta diverse dall'assegnazione di una variabile a un'altra vengono rifiutate, incluso l'accesso agli elementi di iniziazione e raccolta.</p> <p>Stile di distribuzione Amazon Redshift implementato = AUTO.</p>

Origine	Target	Novità, miglioramento o correzione
Oracle	PostgreSQL/Aurora PostgreSQL	<p>Se una parola non riservata in Oracle è riservata in PostgreSQL, è vero quanto segue:</p> <ul style="list-style-type: none"> • Se la parola è citata, mantiene il suo carattere e rimane citata. • Se la parola non è quotata, viene lanciata in maiuscolo e quotata. <p>Implementata la possibilità di utilizzare le funzioni come input per le funzioni LTRIM, RTRIM e TRIM.</p> <p>Le espressioni SELECT DISTINCT, ORDER BY devono essere visualizzate nell'elenco di selezione.</p> <p>Per i parametri del cursore che seguono un parametro con un valore DEFAULT, AWS SCT aggiunge la clausola DEFAULT IS NULL</p> <p>I parametri del cursore OUT di origine vengono convertiti in parametri cursore IN.</p> <p>Variabile pacchetto reimplementato aggiungendo l'opzione "Implementazione logica variabili pacchetto" sotto "Impostazioni di conversione". Le impostazioni disponibili sono: "variabili di sessione" e "oggetti globali plv8". Il valore predefinito è "variabili di sessione".</p> <p>Implementato il supporto pragma AUTONOMOUS_TRANSACTION con dblink e pg_background.</p>
Oracle	Tutti	Implementata visualizzazione SYS_%_TAB_COMMENTS.
Oracle	PostgreSQL	<p>Gli input variabili per i filtri non sono supportati in PostgreSQL. Quando si esegue la conversione da Oracle a PostgreSQL, se viene rilevato un filtro variabile viene segnalata un'eccezione.</p>

Origine	Target	Novità, miglioramento o correzione
Oracle	Amazon Redshift	<p>Implementato codice memorizzato FOR.. LOOP cursore conversione miglioramenti.</p> <p>Implementata invocazione di codice memorizzato di funzione/procedure con parametri predefiniti.</p> <p>Implementata capacità del codice memorizzato per UPDATE con alias senza clausola WHERE.</p> <p>Le funzioni di codice memorizzato implementate preformano casi aggiuntivi con SELECT FROM dual.</p> <p>Implementati i parametri della tabella Table%ROWTYPE e le variabili del pacchetto.</p> <p>Implementato codice memorizzato utilizzato di JAVA e procedure esterne.</p> <p>Implementato il pacchetto Oracle standard nel codice memorizzato.</p>

Modifiche alla release 1.0.640 di Microsoft SQL Server

Nella tabella seguente sono elencate le modifiche di generazione 1.0.640 in cui Microsoft SQL Server è il motore di origine.

Origine	Target	Novità, miglioramento o correzione
Microsoft Azure/ Microsoft SQL Server	PostgreSQL L/Aurora PostgreSQL L/MySQL/ Aurora MySQL	Aggiunto il supporto per gli indici COLUMN STORE.

Origine	Target	Novità, miglioramento o correzione
Microsoft SQL Server	RDS MariaDB 10.4	Aggiunto il supporto RDS MariaDB 10.4 per tutti i fornitori di elaborazione transazionale online (OLTP).
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Aggiunto il supporto per l'attributo indice OPTIMIZE_FOR_SEQUENTIAL_KEY.
Azure/SQL Server	MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Aggiunto il supporto per i tipi di tabella Nodo database e Edge.
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Aggiunto il supporto per TABLES TEMPORAL.

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	Aggiornamento ad Amazon Corretto JDK 11 da JDK 8. Per ulteriori informazioni, inclusi i link per il download, consulta Cos'è Amazon Corretto 11? nella Guida per l'utente di Amazon Corretto 11.
Tutti	Tutti	Sono state aggiunte informazioni al report di valutazione sulle possibili incongruenze nel database dell'utente.
Azure/SQL Server	MySQL/ Aurora MySQL/ Pos tgreSQL/ Aurora PostgreSQL/ MariaDB	Aggiunto il supporto per l'elaborazione DML per SQL Server Graph Architecture.
SQL Server	Aurora PostgreSQL	Aggiunta l'opzione per convertire i parametri senza il prefisso par_.
Azure/SQL Server	MySQL 8.x	Aggiunto il supporto per il vincolo CHECK.
Tutti	SQL Server	<p>SQL Server 2019: aggiunto il supporto per il nuovo attributo indice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: aggiunto il supporto per i tipi di tabella Nodo e Edge Database Graph.</p> <p>SQL Server 2016: aggiunto il supporto per TABLES TEMPORAL.</p>
Tutti	Tutti	Implementata la capacità di sovrascrivere le partizioni fisiche con partizioni virtuali. Gli estrattori di data warehouse estraggono i dati in base alle partizioni virtuali create.

Origine	Target	Novità, miglioramento o correzione
SQL Server	AWS Glue (shell Python)	<p>Miglioramenti della conversione, tra cui:</p> <ul style="list-style-type: none"> • Implementata la conversione delle funzioni incorporate in Python.String. • Implementato EXECUTE ed EXEC nel codice memorizzato. • Implementato utilizzando i tipi di tabella.
Azure/SQL Server	PostgreSQL/Aurora PostgreSQL	Implementato rendendo le procedure \$TMP facoltative.
SQL Server	MySQL/Aurora MySQL	<p>Operazioni aritmetiche estese con date.</p> <p>Emulazione costruzione 'TOP (espressione) CON TIES.</p> <p>Dopo aver chiamato le procedure con il cursore di riferimento, il cursore di riferimento ora si chiude.</p> <p>L'impostazione di un livello di isolamento GLOBAL non è supportata in Aurora MySQL. È possibile modificare solo l'ambito della sessione. Il comportamento predefinito delle transazioni consiste nell'utilizzare REPEATABLE READ e letture consistenti. Potrebbe essere necessario modificare le applicazioni progettate per l'uso con READ COMMITTES. In alternativa, è possibile modificare esplicitamente il valore predefinito in READ COMMITT.</p>

Origine	Target	Novità, miglioramento o correzione
SQL Server	AWS Glue (shell Python)	<p>Le istruzioni di SQL Server producono un set di risultati completo, ma ci sono momenti in cui i risultati sono meglio elaborati una riga alla volta. L'apertura di un cursore su un set di risultati consente di elaborare il set di risultati una riga alla volta. È possibile assegnare un cursore a una variabile o a un parametro con un tipo di dati cursore.</p> <p>Implementato racchiudendo una serie di istruzioni Transact-SQL per il codice archiviato in modo da poter eseguire un gruppo di istruzioni Transact-SQL anche se Python non supporta BEGIN e END as di SQL Server. control-of-flow</p> <p>Le istruzioni SQL Server LABEL e GOTO non sono supportate da AWS Glue Se AWS SCT incontra un'etichetta nel codice, viene saltata. Se AWS SCT incontra un'istruzione GOTO, viene commentata.</p>
SQL Server	Amazon Redshift	<p>Implementata l'elaborazione condizionale delle istruzioni Transact-SQL per il codice memorizzato implementando l'IF... Controllo ELSE.</p> <p>Implementato racchiudendo una serie di istruzioni Transact-SQL per il codice archiviato in modo che un gruppo di istruzioni Transact-SQL possa essere eseguito come blocco. Supporta BEGIN nidificato... Blocchi FINE.</p> <p>Implementato SET e SELECT nel codice memorizzato.</p> <p>È stato implementato CREATE INDEX in Amazon Redshift (che non supporta gli indici) creando una chiave di ordinamento specificata dall'utente nelle tabelle.</p>

Cambiamenti di MySQL versione 1.0.640

La tabella seguente elenca modifiche alla build 1.0.640 in cui MySQL è il motore di origine.

Origine	Target	Novità, miglioramento o correzione
MySQL	PostgreSQL L 12.x	Aggiunto il supporto per le colonne generate.
Tutti	Tutti	Aggiornamento ad Amazon Corretto JDK 11 da JDK 8. Per ulteriori informazioni, inclusi i link per il download, consulta Cos'è Amazon Corretto 11? nella Guida per l'utente di Amazon Corretto 11.
Tutti	Tutti	Sono state aggiunte informazioni al report di valutazione sulle possibili incongruenze nel database dell'utente.
MySQL	PostgreSQL L/Aurora PostgreSQL L 11.	<p>Aggiunto il supporto per le seguenti lingue:</p> <ul style="list-style-type: none"> • Transazioni incorporate all'interno delle stored procedure SQL. • La possibilità di CALL SQL stored procedure. • Possibilità di creare stored procedure SQL.
Tutti	SQL Server	<p>SQL Server 2019: aggiunto il supporto per il nuovo attributo indice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: aggiunto il supporto per i tipi di tabella Nodo e Edge Database Graph.</p> <p>SQL Server 2016: aggiunto il supporto per TABLES TEMPORAL.</p>
Tutti	Tutti	Implementata la capacità di sovrascrivere le partizioni fisiche con partizioni virtuali. Gli estrattori di data warehouse estraggono i dati in base alle partizioni virtuali create.

Modifiche a PostgreSQL nella release 1.0.640

Nella tabella seguente sono elencate le modifiche di build 1.0.640 in cui PostgreSQL è il motore di origine.

Origine	Target	Novità, miglioramento o correzione
PostgreSQL	MySQL 8.x	<p>MySQL ora supporta la creazione di parti chiave di indice funzionali che indicizzano valori di espressione anziché valori di colonna. Le parti chiave funzionali consentono l'indicizzazione di valori che non possono essere indicizzati altrimenti.</p> <p>MySQL ora supporta Now CTE e CTE Recursive.</p>
Tutti	Tutti	<p>Aggiornamento ad Amazon Corretto JDK 11 da JDK 8. Per ulteriori informazioni, inclusi i link per il download, consulta Cos'è Amazon Corretto 11? nella Guida per l'utente di Amazon Corretto 11.</p>
Tutti	Tutti	<p>Sono state aggiunte informazioni al report di valutazione sulle possibili incongruenze nel database dell'utente.</p>
PostgreSQL 11.x	PostgreSQL/Aurora PostgreSQL 11.	<p>Aggiunto il supporto per le seguenti lingue:</p> <ul style="list-style-type: none"> • Transazioni incorporate all'interno delle stored procedure SQL. • La possibilità di CALL SQL stored procedure. • Possibilità di creare stored procedure SQL.
PostgreSQL	MySQL 8.x	<p>Aggiunto il supporto MySQL per gli indici decrescenti. DESC in una definizione di indice non viene più ignorato, ma causa la memorizzazione dei valori chiave in ordine decrescente.</p> <p>Aggiunto supporto MySQL all'uso di espressioni come valori predefiniti nelle specifiche del tipo di dati, comprese le espressioni come valori predefiniti per i tipi di dati BLOB, TEXT, GEOMETRY e JSON.</p> <p>Diverse funzioni di aggregazione esistenti ora possono essere utilizzate come funzioni di finestra.</p> <ul style="list-style-type: none"> • AVG() • BIT_AND() • BIT_OR()

Origine	Target	Novità, miglioramento o correzione
		<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 supporta le funzioni di finestra che, per ogni riga di una query, eseguono un calcolo utilizzando le righe relative a tale riga.</p> <ul style="list-style-type: none"> • CUME_DIST() • DENSE_RANK() • FIRST_VALUE() • LAG() • LAST_VALUE() • LEAD() • NTH_VALUE() • NTILE() • PERCENT_RANK() • RANK() • ROW_NUMBER()

Origine	Target	Novità, miglioramento o correzione
PostgreSQL	MySQL 8.x	Aggiunto il supporto per il vincolo CHECK.
Tutti	SQL Server	<p>SQL Server 2019: aggiunto il supporto per il nuovo attributo indice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: aggiunto il supporto per i tipi di tabella Nodo e Edge Database Graph.</p> <p>SQL Server 2016: aggiunto il supporto per TABLES TEMPORAL.</p>
Tutti	Tutti	Implementata la capacità di sovrascrivere le partizioni fisiche con partizioni virtuali. Gli estrattori di data warehouse estraggono i dati in base alle partizioni virtuali create.
PostgreSQL/Aurora PostgreSQL	Tutti	<p>Aggiunta emulazione sysindexes vista sistema.</p> <p>Se in una routine è presente un'istruzione SELECT senza specificare INTO, viene creato il parametro INOUT p_refcur di tipo refcursor per una routine sulla destinazione.</p>

Modifiche alla versione 1.0.640 Db2 LUW

La tabella seguente elenca modifiche alla build 1.0.640 in cui DB2 LUW è il motore di origine.

Origine	Target	Novità, miglioramento o correzione
DB2 LUW	RDS MariaDB 10.4	Aggiunto il supporto RDS MariaDB 10.4 per tutti i fornitori di elaborazione transazionale online (OLTP).
Tutti	Tutti	Aggiornamento ad Amazon Corretto JDK 11 da JDK 8. Per ulteriori informazioni, inclusi i link per il download, consulta Cos'è Amazon Corretto 11? nella Guida per l'utente di Amazon Corretto 11.

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	Sono state aggiunte informazioni al report di valutazione sulle possibili incongruenze nel database dell'utente.
DB2 LUW	MySQL 8.0.17	Aggiunto il supporto del vincolo CHECK.
Tutti	SQL Server	<p>SQL Server 2019: aggiunto il supporto per il nuovo attributo indice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: aggiunto il supporto per i tipi di tabella Nodo e Edge Database Graph.</p> <p>SQL Server 2016: aggiunto il supporto per TABLES TEMPORAL.</p>
Tutti	Tutti	Implementata la capacità di sovrascrivere le partizioni fisiche con partizioni virtuali. Gli estrattori di data warehouse estraggono i dati in base alle partizioni virtuali create.

Modifiche alla versione 1.0.640 di Teradata

La tabella seguente elenca le modifiche apportate alla versione 1.0.640 dei motori di origine di Teradata.

Origine	Target	Novità, miglioramento o correzione
Teradata	Amazon Redshift	<p>Aggiunto il supporto per le istruzioni MERGE e QUALIFY.</p> <p>Rimossa la clausola LOCKING ROWS FOR ACCESS dalle istruzioni Teradata.</p> <p>Aggiunto il supporto per la funzione CAST.</p>
Tutti	Tutti	Aggiornamento ad Amazon Corretto JDK 11 da JDK 8. Per ulteriori informazioni, inclusi i link per il download, consulta Cos'è Amazon Corretto 11? nella Guida per l'utente di Amazon Corretto 11.

Origine	Target	Novità, miglioramento o correzione
Teradata	Teradata	Miglioramenti implementati in REGEXP_INSTR () e REGEXP_SUBSTR ().
Tutti	Tutti	Sono state aggiunte informazioni al report di valutazione sulle possibili incongruenze nel database dell'utente.
Tutti	SQL Server	<p>SQL Server 2019: aggiunto il supporto per il nuovo attributo indice OPTIMIZE_FOR_SEQUENTIAL_KEY.</p> <p>SQL Server 2017: aggiunto il supporto per i tipi di tabella Nodo e Edge Database Graph.</p> <p>SQL Server 2016: aggiunto il supporto per TABLES TEMPORAL.</p>
Teradata	Tutti	Aggiunto il supporto per REGEXP_INSTR () e REGEXP_SUBSTR ().
Tutti	Tutti	Implementata la capacità di sovrascrivere le partizioni fisiche con partizioni virtuali. Gli estrattori di data warehouse estraggono i dati in base alle partizioni virtuali create.
Teradata	Amazon Redshift	<p>Implementata la possibilità di salvare SQL dell'albero di origine in file singolo o in più file per fase utilizzando le impostazioni in Impostazioni progetto, Salva come SQL e Applica, Elenco a discesa: File singolo/File multipli.</p> <p>Miglioramenti implementati nelle conversioni di visualizzazioni e procedure.</p>
Teradata	Tutti	Aggiunto il supporto per Teradata versione 16.20

Modifiche alla versione 1.0.640 per altri motori

La tabella seguente elenca le modifiche alla build 1.0.640 in altri motori di origine.

Origine	Target	Novità, miglioramento o correzione
Sybase	RDS MariaDB 10.4	Aggiunto il supporto RDS MariaDB 10.4 per tutti i fornitori di elaborazione transazionale online (OLTP).
SAP ASE	MariaDB	Implementato quanto segue: <ul style="list-style-type: none"> • MariaDB 10.4 • Istruzione EXECUTE IMMEDIATE • Definizioni DEFAULT • Supporto vincolo CHECK
SAP ASE	PostgreSQL L 12.x	Aggiunto il supporto per le colonne generate.
Tutti	Tutti	Aggiornamento ad Amazon Corretto JDK 11 da JDK 8. Per ulteriori informazioni, inclusi i link per il download, consulta Cos'è Amazon Corretto 11? nella Guida per l'utente di Amazon Corretto 11.
Tutti	Tutti	Sono state aggiunte informazioni al report di valutazione sulle possibili incongruenze nel database dell'utente.
SAP ASE	MySQL 8.0.17	Aggiunto il supporto del vincolo CHECK.
Tutti	SQL Server	SQL Server 2019: aggiunto il supporto per il nuovo attributo indice OPTIMIZE_FOR_SEQUENTIAL_KEY. SQL Server 2017: aggiunto il supporto per i tipi di tabella Nodo e Edge Database Graph. SQL Server 2016: aggiunto il supporto per TABLES TEMPORAL.
Vertica	Amazon Redshift	Aggiunto il supporto per lo stile di distribuzione = AUTO.

Origine	Target	Novità, miglioramento o correzione
Tutti	Tutti	Implementata la capacità di sovrascrivere le partizioni fisiche con partizioni virtuali. Gli estrattori di data warehouse estraggono i dati in base alle partizioni virtuali create.
Amazon Redshift	Amazon Redshift	Le funzioni incorporate non supportate nelle istruzioni DML vengono sostituite con NULL come segnaposto.
Sybase	PostgreSQL	Aggiunto il supporto per le funzioni native.
SAP ASE	MySQL/ Aurora MySQL	Il livello di isolamento predefinito per Aurora MySQL è REPEATABLE READ. L'impostazione di un livello di isolamento GLOBAL non è supportata in Aurora MySQL. È possibile modificare solo l'ambito della sessione. Il comportamento predefinito delle transazioni consiste nell'utilizzare REPEATABLE READ e letture consistenti. Potrebbe essere necessario modificare le applicazioni progettate per l'esecuzione con READ COMMITTED. Oppure puoi modificare esplicitamente il valore predefinito in READ COMMITTED.
SAP ASE	PostgreSQL	Aggiunto il supporto per la funzione CONVERT (ottimistica) senza il pacchetto di estensione.
SAP ASE	Tutti	<p>Aggiunta emulazione sysindexes vista sistema.</p> <p>Se in una routine è presente un'istruzione SELECT senza specificare INTO, viene creato il parametro INOUT p_refcur di tipo refcursor per una routine sulla destinazione.</p>
Greenplum	Amazon Redshift	Implementato CREATE TABLE TEMPORARY come segue:

Cronologia dei documenti

La tabella seguente descrive le modifiche importanti alla guida per l'utente AWS Schema Conversion Tool (AWS SCT) dopo gennaio 2018.

Puoi abbonarti a un feed RSS per ricevere le notifiche sugli aggiornamenti di questa documentazione.

Modifica	Descrizione	Data
AWS SCTbuild #1 .0.672	La build 1.0.672 fornisce il supporto di Amazon RDS per PostgreSQL 15 come destinazione e Microsoft SQL Server versione 2022 come origine. Aggiunge inoltre il supporto per le nuove funzionalità di Amazon Redshift nel codice convertito, implementa diversi miglioramenti di conversione per il sorgente IBM Db2 for z/OS e risolve una serie di problemi di conversione.	8 maggio 2023
AWS SCTbuild #1 .0.671	La build 1.0.671 fornisce il supporto per le migrazioni da Apache Oozie a AWS Step Functions. Aggiunge inoltre il supporto BigQuery come fonte per il processo di valutazione multiserver. Inoltre, aggiunge nuove impostazioni di conversione per IBM Db2 for z/OS come fonte e risolve una serie di problemi di conversione.	8 marzo 2023

[AWS SCTbuild #1 .0.670](#)

La build 1.0.670 fornisce supporto per le migrazioni da Hadoop ad Amazon EMR. Aggiunge anche il supporto per Azure Synapse Analytics come fonte per il processo di valutazione multiserver. Inoltre, migliora la conversione del codice SQL incorporato nelle applicazioni Java e risolve una serie di problemi di conversione.

23 gennaio 2023

[AWS SCTbuild #1 .0.669](#)

La build 1.0.669 implementa il supporto del partizionamento nativo per la migrazione dei dati dai data warehouse Oracle. Inoltre, migliora il processo di valutazione multiserver, aggiunge nuove funzionalità negli agenti di estrazione dei dati e risolve una serie di problemi di conversione.

19 dicembre 2022

[AWS SCTbuild #1 .0.668](#)

Build 1.0.668 implementa il partizionamento virtuale automatico per la migrazione dei dati dai database Greenplum e aggiunge il supporto per la migrazione dei dati dai database Snowflake ad Amazon Redshift. Inoltre, migliora la conversione del codice SQL incorporato nelle applicazioni C# e risolve una serie di problemi di conversione.

16 novembre 2022

[AWS SCTbuild #1 .0.667](#)

La build 1.0.667 fornisce il supporto per il motore di estrazione, trasformazione e caricamento (ETL) di Informatica come fonte di migrazione. Inoltre, aggiorna la versione del pacchetto di estensione, aumenta la versione minima del driver supportata per Amazon Redshift e risolve una serie di problemi di conversione.

13 ottobre 2022

[AWS SCTbuild #1 .0.666](#)

La build 1.0.666 migliora la conversione delle applicazioni Java aggiungendo il supporto per il framework . MyBatis Inoltre, aggiunge nuove funzioni nei pacchetti di estensioni, migliora il caricatore e dei metadati di origine e risolve una serie di problemi di conversione.

20 settembre 2022

[AWS SCTbuild #1 .0.665](#)

La build 1.0.665 fornisce il supporto di BigQuery come fonte di migrazione. Implementa anche il supporto della nuova versione del file di configurazione delle funzionalità di Babelfish. Inoltre, migliora la conversione dei data warehouse in Amazon Redshift e risolve una serie di problemi di conversione.

29 agosto 2022

[AWS SCTbuild #1 .0.664](#)

La build 1.0.664 fornisce il supporto di Amazon Redshift Serverless come origine o destinazione di migrazione e. Implementa anche il bilanciamento automatico della memoria nelle attività di estrazione dei dati e corregge un errore che AWS SCT impediva la connessione ai dispositivi. AWS Snowball Inoltre, aggiunge la possibilità di modificare il confronto delle colonne nelle regole di migrazione, migliora l'interfaccia utente e risolve una serie di problemi di conversione.

14 luglio 2022

[AWS SCTbuild #1 .0.663](#)

La build 1.0.663 aggiunge il supporto per Babelfish for Aurora PostgreSQL 1.2.0 e migliora le funzionalità dei report di valutazione multiserver. Inoltre, aggiunge nuove funzionalità nelle regole di migrazione, corregge due errori del caricatore e risolve una serie di problemi di conversione.

20 giugno 2022

[AWS SCTbuild #1 .0.662](#)

La build 1.0.662 implementa la conversione del codice SQL nelle applicazioni C# e migliora il flusso di lavoro dei report di valutazione multiserver. Inoltre, aggiunge diversi miglioramenti alla conversione e risolve una serie di problemi di conversione.

19 maggio 2022

[AWS SCTbuild #1 .0.661](#)

La build 1.0.661 fornisce il supporto di IBM Db2 for z/OS come fonte di migrazione. Aggiunge inoltre il supporto per la conversione degli script di estrazione, trasformazione e caricamento (ETL) AWS Glue Studio e risolve una serie di problemi di conversione.

21 aprile 2022

[AWS SCTbuild #1 .0.660](#)

La build 1.0.660 fornisce il supporto della versione principale 14 di PostgreSQL e MariaDB 10.6 come obiettivi di migrazione. Aggiunge inoltre il supporto per la conversione degli indici Oracle per le viste materializzate e risolve una serie di problemi di conversione.

21 marzo 2022

[AWS SCTbuild #1 .0.659](#)

La build 1.0.659 fornisce il supporto della versione principale 13 di PostgreSQL su Aurora PostgreSQL compatibile con Edition come obiettivo di migrazione e. Implementa la conversione del codice SQL nelle applicazioni C#, aggiunge il supporto di Oracle Unified Auditing e risolve una serie di problemi di conversione.

21 febbraio 2022

[AWS SCTbuild #1 .0.658](#)

La build 1.0.658 fornisce l'integrazione AWS Secrets Manager e aggiunge il supporto della piattaforma di database di destinazione virtuale Amazon Redshift. Aggiunge anche una serie di miglioramenti alla conversione e correzioni di bug.

20 gennaio 2022

[AWS SCTbuild #1 .0.657](#)

La build 1.0.657 migliora la conversione da Microsoft SQL Server a Aurora PostgreSQL compatibile con Edition, Amazon RDS per PostgreSQL e altre destinazioni di migrazione. Aggiunge inoltre una serie di miglioramenti dell'interfaccia utente e correzioni di bug.

20 dicembre 2021

[AWS SCTbuild #1 .0.656](#)

La build 1.0.656 fornisce il supporto di più database di origine e destinazione in un unico progetto. Aggiunge inoltre una strategia di conversione, ottimizzazione e miglioramenti generali e una serie di correzioni di bug.

22 novembre 2021

[AWS SCTbuild #1 .0.655](#)

La build 1.0.655 implementa la conversione degli script di FastExport lavoro Teradata in Amazon Redshift RSQL e aumenta la versione minima del driver supportata per Greenplum a 42.2.5. Aggiunge anche una serie di miglioramenti e correzioni di bug.

18 ottobre 2021

[AWS SCTbuild #1 .0.654](#)

La build 1.0.654 implementa la conversione degli script Shell, Teradata e Teradata Basic Teradata FastLoad Query (BTEQ) in Amazon Redshift RSQL. Inoltre, risolve una serie di problemi di conversione e aggiunge una serie di miglioramenti e correzioni di bug.

16 settembre 2021

[AWS SCTbuild #1 .0.653](#)

La build 1.0.653 implementa la conversione di SQL dinamico creato in funzioni o procedure chiamate. Inoltre, migliora la conversione delle routine crittografate e aggiunge una serie di miglioramenti e correzioni di bug.

10 agosto 2021

[AWS SCTbuild #1 .0.652](#)

La build 1.0.652 implementa la modalità di comando dello script nell'interfaccia della riga di comando e implementa le regole di ottimizzazione dello schema. Aggiunge inoltre una serie di miglioramenti delle conversioni e delle prestazioni e correzioni di bug.

30 giugno 2021

[AWS SCTbuild #1 .0.651](#)

La build 1.0.651 aggiunge una serie di miglioramenti e correzioni di errori. Fornisce inoltre l'accesso alla copia iniziale del riferimento AWS Schema Conversion Tool CLI.

4 giugno 2021

[AWS SCTbuild #1 .0.650](#)

La build 1.0.650 implementa il supporto di Amazon RDS per PostgreSQL 13 come database di destinazione, aggiorna gli agenti di estrazione. Inoltre, aggiorna la versione minima del driver JDBC supportata per Microsoft SQL Server, Azure e Azure Synapse. Inoltre, aggiunge una serie di miglioramenti alla conversione e correzioni di bug.

30 aprile 2021

[AWS SCTbuild #1 .0.649](#)

La build 1.0.649 implementa il supporto di MariaDB 10.5 come database di destinazione e implementa miglioramenti delle funzioni per la conversione delle funzioni integrate di Oracle. Aggiunge inoltre una serie di miglioramenti delle conversioni e delle prestazioni e correzioni di bug.

29 marzo 2021

[AWS SCTbuild #1 .0.648](#)

La build 1.0.648 aggiunge una serie di miglioramenti alla conversione e correzioni di errori.

22 febbraio 2021

AWS SCTbuild #1 .0.647	La build 1.0.647 aggiunge il supporto della funzionalità Database Mail su Amazon RDS, implementa il caricamento e la conversione dei commenti sugli oggetti di storage. Aggiunge inoltre AWS SCT Data Migration Service Assessor e AWS SCT Wizard e implementa l'interfaccia utente del filtro ad albero. Inoltre, aggiunge una sezione riprogettata nel rapporto di valutazione e una serie di miglioramenti e correzioni di bug.	15 gennaio 2021
AWS SCTbuild #1 .0.646	La build 1.0.646 aggiunge il supporto per i tipi di dati INTERVAL, le colonne di identità e la conversione dei cursori e aggiunge una serie di miglioramenti e correzioni di errori.	28 dicembre 2020
AWS SCTbuild #1 .0.645	La build 1.0.645 aggiunge il supporto per ETL SSIS alla AWS Glue conversione e una serie di miglioramenti e correzioni di bug.	16 Novembre 2020
AWS SCTbuild #1 .0.643-1.0.644	La build 1.0.644 aggiunge una serie di miglioramenti alla conversione, alle prestazioni e all'interfaccia utente e correzioni di bug.	14 ottobre 2020

AWS SCTbuild #1 .0.642	La build 1.0.642 implementa la conversione dei pacchetti ETL da Microsoft SQL Server Integration Services AWS Glue e aggiunge una serie di miglioramenti e correzioni di bug.	28 agosto 2020
AWS SCTbuild #1 .0.641	È stato aggiunto il supporto SSL per gli estrattori di dati. Build include anche una serie di miglioramenti e correzioni.	17 luglio 2020
AWS SCTcostruisce #1 .0.633-1.0.640	Aggiornato da JDK 8 ad Amazon Corretto JDK 11. Aggiunte tabelle che identificano altri aggiornamenti, modifiche e correzioni.	22 giugno 2020
Disponibilità di AWS WQF	AWS SCT non fornisce più lo strumento AWS Workload Qualification Framework (AWS WQF) per il download.	19 giugno 2020
AWS SCTcostruisce #1 .0.632	SCT UI - Aggiunta una nuova scheda per mostrare gli errori che si verificano quando si applicano gli script. È ora possibile salvare la struttura di origine come SQL durante la conversione da SAP ASE. Miglioramenti per le conversioni in PostgreSQL o Aurora PostgreSQL o Redshift.	19 novembre 2019

[AWS SCTbuild #1 .0.631 e
#1 .0.630 \(combinate\)](#)

Supporto migliore per i ROWID in Oracle e per gli oggetti di sistema in Microsoft SQL Server e SAP ASE. Gestione migliorata per gli specificatori mancanti degli schemi di SQL Server. Supporto migliorato per le conversioni da Greenplum a Redshift. Supporto migliorato per la conversione del codice archiviato quando si passa ad Amazon Redshift, MariaDB, MySQL e PostgreSQL.

30 settembre 2019

[AWS SCTbuild #1 .0.629](#)

Supporto per le procedure memorizzate per le conversioni da Netezza. Supporto migliorato per le conversioni in Amazon Redshift, DynamoDB, MySQL e PostgreSQL. Aggiunta del supporto per SAP ASE 12.5 come origine.

20 agosto 2019

[AWS SCTbuild #1 .0.628](#)

Supporto per l'emulazione del servizio per le conversioni da DB2, SQL Server e Oracle. Miglioramenti per le conversioni ad Amazon Redshift, incluso un maggiore supporto per cursori e procedure archiviate.

22 giugno 2019

[AWS SCTbuild #1 .0.627](#)

Supporto per le conversioni da SQL Server a procedure archiviate in Amazon Redshift. Miglioramenti per la conversione in PostgreSQL 11 e MySQL 8.0.

31 maggio 2019

[AWS SCTbuild #1 .0.626](#)

PostgreSQL 11 e MySQL 8.0 sono ora supportati come obiettivi. SAP ASE 15.5 è ora supportato come origine.

26 aprile 2019

[AWS SCTbuild #1 .0.625](#)

Gli aggiornamenti includono la possibilità di convertire Teradata BTEQ in AWS Glue, il supporto per le conversioni a MariaDB 10.3 con supporto della modalità di compatibilità Oracle, il supporto per SAP ASE 15.7 e la sostituzione dei servizi per emulare le funzionalità mancanti.

25 marzo 2019

[AWS SCTbuild #1 .0.624](#)

Gli aggiornamenti includono la possibilità di convertire Oracle ETL in Amazon RDS per AWS Glue MariaDB e il supporto per le conversioni da Microsoft SQL Server, Oracle e IBM Db2 LUW. È stato inoltre aggiunto il supporto per conversioni da SAP ASE a RDS for MySQL e Amazon Aurora con compatibilità MySQL. È stato inoltre aggiunto il supporto per l'estensione Oracle durante la conversione da Oracle a PostgreSQL.

22 febbraio 2019

[AWS SCTbuild #1 .0.623](#)

Gli aggiornamenti includono la possibilità di convertire i database SAP ASE, nonché gli script T-SQL, DML e DDL in codice o componenti equivalenti. Sono state inoltre aggiunte le emulazioni Oracle e Microsoft SQL Server per migliorare le conversioni.

25 gennaio 2019

[AWS SCTbuild #1 .0.622](#)

Gli aggiornamenti includono Workload Qualification Framework, che analizza il carico di lavoro per un'intera migrazione, incluse le modifiche al database e alle app.

20 dicembre 2018

AWS SCTbuild #1 .0.621	Gli aggiornamenti includono il supporto per Aurora PostgreSQL 10 come destinazione e la possibilità di eseguire la migrazione da Netezza utilizzando le opzioni della tabella esterna.	21 novembre 2018
AWS SCTbuild #1 .0.620	Gli aggiornamenti includono il salvataggio degli script SQL e il supporto per i cursori globali Oracle durante la migrazione a MySQL.	22 ottobre 2018
AWS SCTbuild #1 .0.619	Gli aggiornamenti includono il supporto per la migrazione da Apache Cassandra a DynamoDB e il supporto per Vertica 9 come origine.	20 settembre 2018
AWS SCTbuild #1 .0.618	Gli aggiornamenti includono report di valutazione ampliati, il supporto per la conversione di ROWID Oracle e il supporto per le tabelle definite dall'utente SQL Server.	24 agosto 2018
AWS SCTbuild #1 .0.617	Gli aggiornamenti includono report di valutazione ampliati, il supporto per la conversione di ROWID Oracle e il supporto per le tabelle definite dall'utente SQL Server.	24 luglio 2018

AWS SCTbuild #1 .0.616	Gli aggiornamenti includono il supporto per RDS durante la conversione da Oracle ad Amazon RDS per Oracle, la conversione di oggetti di pianificazione Oracle, il supporto per processi Oracle, il partizionamento e Db2 LUW versione 10.1.	26 giugno 2018
AWS SCTbuild #1 .0.615	Gli aggiornamenti includono il supporto per le istruzioni GOTO da SQL Server a PostgreSQL, il partizionamento PostgreSQL 10 e Db2 LUW versione 10.1.	24 maggio 2018
AWS SCTbuild #1 .0.614	Gli aggiornamenti includono il supporto per i link da Oracle a database Oracle, funzioni Inline da SQL Server a PostgreSQL e l'emulazione di oggetti di sistema Oracle.	25 aprile 2018
AWS SCTbuild #1 .0.613	Gli aggiornamenti includono il supporto per Db2 LUW, la conversione di file SQL* Plus e l'autenticazione Windows SQL Server.	28 marzo 2018
AWS SCTbuild #1 .0.612	Gli aggiornamenti includono il supporto per la mappatura personalizzata dei tipi di dati, il confronto dello schema per Oracle 10 e la conversione da PostgreSQL a Oracle di variabili globali.	22 febbraio 2018

[AWS SCTbuild #1 .0.611](#)

Gli aggiornamenti includono il supporto per le istruzioni dinamiche da Oracle a PostgreSQL, l'apertura del file di log selezionando un messaggio di errore e la possibilità di nascondere schemi nella visualizzazione ad albero.

23 gennaio 2018

Aggiornamenti precedenti

La tabella seguente descrive le modifiche importanti alla guida per l'utente AWS Schema Conversion Tool (AWS SCT) prima di gennaio 2018.

Versione	Modifica	Descrizione	Data della modifica
1.0.608	Supporto per endpoint FIPS per Amazon S3	Ora puoi richiedere di connetterti AWS SCT ad Amazon S3 e Amazon Redshift utilizzando gli endpoint FIPS per soddisfare i requisiti di sicurezza del Federal Information Processing Standard. Per ulteriori informazioni, consulta ArchiviazioneAWS delle credenziali .	17 Novembre 2017
1.0.607	Supporto per endpoint FIPS per Amazon S3	Ora puoi richiedere di connetterti AWS SCT ad Amazon S3 e Amazon Redshift utilizzando gli endpoint FIPS per soddisfare i requisiti di sicurezza del Federal Information Processing Standard. Per ulteriori informazioni, consulta ArchiviazioneAWS delle credenziali .	30 ottobre 2017
1.0.607	Le attività di estrazione dei	Quando crei le attività di estrazione dei dati, puoi scegliere di ignorare gli oggetti di grandi	30 ottobre 2017

Versione	Modifica	Descrizione	Data della modifica
	dati possono ignorare i LOB	dimensioni (LOB) per ridurre la quantità di dati che estrai. Per ulteriori informazioni, consulta Creazione, esecuzione e monitoraggio di un'attività di estrazione dei dati AWS SCT .	
1.0.605	Accesso al log dell'attività dell'agente di estrazione dei dati	Da questo momento puoi accedere al log dell'attività dell'agente di estrazione dei dati da un comodo collegamento nell'interfaccia utente di AWS Schema Conversion Tool. Per ulteriori informazioni, consulta Creazione, esecuzione e monitoraggio di un'attività di estrazione dei dati AWS SCT .	28 agosto 2017
1.0.604	Miglioramenti Converter	Il motore AWS Schema Conversion Tool è stato migliorato per offrire conversioni migliorate per migrazioni eterogenee.	24 giugno 2017
1.0.603	Gli agenti di estrazione dei dati supportano i filtri	Da questo momento puoi filtrare i dati che gli agenti di estrazione estraggono dal data warehouse. Per ulteriori informazioni, consulta Creazione di regole di migrazione dei dati in AWS SCT .	16 giugno 2017
1.0.603	AWS SCT supporta versioni di data warehouse aggiuntivi	Ora puoi utilizzarlo AWS Schema Conversion Tool per convertire gli schemi Teradata 13 e Oracle Data Warehouse 10 in schemi Amazon Redshift equivalenti. Per ulteriori informazioni, consulta Conversione di schemi di data warehouse in Amazon Redshift utilizzando AWS SCT .	16 giugno 2017

Versione	Modifica	Descrizione	Data della modifica
1.0.602	Gli agenti di estrazione dei dati supportano data warehouse aggiuntivi	Da questo momento puoi utilizzare gli agenti di estrazione dei dati per estrarre i dati dal data warehouse di Microsoft SQL Server. Per ulteriori informazioni, consulta Migrazione e dei dati da un data warehouse locale ad Amazon Redshift .	11 maggio 2017
1.0.602	Gli agenti di estrazione dei dati possono copiare i dati su Amazon Redshift	Da questo momento gli agenti di estrazione dei dati dispongono di tre modalità di upload. Ora puoi specificare se estrarre semplicemente i dati, estrarli e caricarli semplicemente su Amazon S3 oppure estrarre, caricare e copiare i dati direttamente in Amazon Redshift. Per ulteriori informazioni, consulta Creazione, esecuzione e monitoraggio di un'attività di estrazione dei dati AWS SCT .	11 maggio 2017
1.0.601	AWS SCT supporta data warehouse aggiuntivi	Ora puoi utilizzare il AWS Schema Conversion Tool per convertire gli schemi Vertica e Microsoft SQL Server in schemi Amazon Redshift equivalenti. Per ulteriori informazioni, consulta Conversione di schemi di data warehouse in Amazon Redshift utilizzando AWS SCT .	18 Aprile 2017
1.0.601	Gli agenti di estrazione dei dati supportano data warehouse aggiuntivi	Da questo momento puoi utilizzare gli agenti di estrazione dei dati per estrarre i dati dal data warehouse di Greenplum, Netezza, and Vertica. Per ulteriori informazioni, consulta Migrazione dei dati da un data warehouse locale ad Amazon Redshift .	18 Aprile 2017

Versione	Modifica	Descrizione	Data della modifica
1.0.601	Gli agenti di estrazione dei dati supportano sistemi operativi aggiuntivi	Da questo momento puoi installare agenti di estrazione dei dati su computer che eseguono i sistemi operativi macOS e Microsoft Windows. Per ulteriori informazioni, consulta Installazione degli agenti di estrazione .	18 Aprile 2017
1.0.601	Gli agenti di estrazione dei dati vengono caricati automaticamente su Amazon S3	Gli agenti di estrazione dei dati ora caricano automaticamente i dati estratti su Amazon S3. Per ulteriori informazioni, consulta Output dell'attività di estrazione dei dati .	18 Aprile 2017
1.0.600	Agenti di estrazione dei dati	Ora puoi installare agenti di estrazione dei dati che estraggono i dati dal tuo data warehouse e li preparano per l'uso con Amazon Redshift. Puoi utilizzare AWS Schema Conversion Tool per registrare gli agenti e creare per loro attività di estrazione dei dati. Per ulteriori informazioni, consulta Migrazione dei dati da un data warehouse locale ad Amazon Redshift .	16 febbraio 2017
1.0.600	Suggerimenti dei clienti	Ora puoi fornire un feedback su AWS Schema Conversion Tool. Puoi archiviare un report sui bug, inviare una richiesta di funzionalità oppure fornire informazioni generali. Per ulteriori informazioni, consulta Fornire feedback .	16 febbraio 2017

Versione	Modifica	Descrizione	Data della modifica
1.0.502	Integrazione con AWS DMS	Da questo momento puoi utilizzare AWS Schema Conversion Tool per creare endpoint e attività di AWS DMS. Puoi eseguire e monitorare le attività da AWS SCT. Per ulteriori informazioni, consulta Uso di AWS SCT con AWS DMS .	20 dicembre 2016
1.0.502	Amazon Aurora con compatibilità PostgreSQL come database di destinazione	Da questo momento AWS Schema Conversion Tool supporta Amazon Aurora con compatibilità PostgreSQL come database di destinazione. Per ulteriori informazioni, consulta Conversione di schemi di database utilizzando AWS SCT .	20 dicembre 2016
1.0.502	Supporto dei profili	Da questo momento puoi archiviare diversi profili in AWS Schema Conversion Tool e passare facilmente da uno all'altro. Per ulteriori informazioni, consulta Archiviazione dei profili AWS di servizio in AWS SCT .	20 dicembre 2016
1.0.501	Supporto per Greenplum Database e Netezza	Ora puoi utilizzarlo AWS Schema Conversion Tool per convertire gli schemi del tuo data warehouse da Greenplum Database e Netezza ad Amazon Redshift. Per ulteriori informazioni, consulta Conversione di schemi di data warehouse in Amazon Redshift utilizzando AWS SCT .	17 Novembre 2016
1.0.501	Ottimizzazione di Redshift	Ora puoi utilizzarlo AWS Schema Conversion Tool per ottimizzare i tuoi database Amazon Redshift. Per ulteriori informazioni, consulta Ottimizzazione di Amazon Redshift utilizzando AWS SCT .	17 Novembre 2016

Versione	Modifica	Descrizione	Data della modifica
1.0.500	Regole di mappatura	Da questo momento, prima di convertire lo schema con AWS Schema Conversion Tool, puoi impostare regole in grado di modificare il tipo di dati delle colonne, spostare oggetti da uno schema a un altro e modificare il nome degli oggetti. Per ulteriori informazioni, consulta Creazione di regole di migrazione in AWS SCT .	4 Ottobre 2016
1.0.500	Passaggio al cloud	Ora puoi utilizzarlo AWS Schema Conversion Tool per copiare lo schema del tuo database locale esistente su un'istanza a database Amazon RDS che esegue lo stesso motore. Puoi utilizzare questa caratteristica per analizzare il potenziale risparmio sui costi del passaggio al cloud e della modifica del tipo di licenza. Per ulteriori informazioni, consulta Creazione di report di valutazione della migrazione con AWS SCT .	4 Ottobre 2016
1.0.400	Conversioni dello schema del data warehouse	Ora puoi utilizzarlo AWS Schema Conversion Tool per convertire gli schemi del tuo data warehouse da Oracle e Teradata ad Amazon Redshift. Per ulteriori informazioni, consulta Conversione di schemi di data warehouse in Amazon Redshift utilizzando AWS SCT .	13 luglio 2016

Versione	Modifica	Descrizione	Data della modifica
1.0.400	Conversioni SQL delle applicazioni	Da questo momento puoi utilizzare AWS Schema Conversion Tool per convertire SQL in codice C++, C#, Java o altro tipo di codice dell'applicazione. Per ulteriori informazioni, consulta Conversione del codice SQL dell'applicazione utilizzando AWS SCT .	13 luglio 2016
1.0.400	Nuova caratteristica	AWS Schema Conversion Tool Ora contiene un pacchetto di estensioni e una procedura guidata per aiutarti a installare, creare e configurare AWS Lambda funzioni e librerie Python per fornire e-mail, pianificazione dei lavori e altre funzionalità. Per ulteriori informazioni, consultare Utilizzo delle AWS Lambda funzioni del pacchetto di AWS SCT estensione e Utilizzo di librerie personalizzate per i pacchetti di AWS SCT estensione .	13 luglio 2016
1.0.301	Supporto per SSL	Da questo momento puoi utilizzare Secure Sockets Layer (SSL) per connetterti al database di origine quando utilizzi AWS Schema Conversion Tool.	19 maggio 2016
1.0.203	Nuova caratteristica	Aggiunge il supporto per MySQL e PostgreSQL come database di origine per le conversioni.	11 aprile 2016

Versione	Modifica	Descrizione	Data della modifica
1.0.202	Versione di manutenzione	Aggiunge il supporto per modificare l'SQL convertito che è stato generato per il motore di database di destinazione. Aggiunge funzionalità di selezione migliorate nelle visualizzazioni ad albero delle istanze database di origine e di destinazione. Aggiunge il supporto per la connessione a un database di origine Oracle utilizzando nomi del Transparent Network Substrate (TNS).	2 marzo 2016
1.0.200	Versione di manutenzione	Aggiunge il supporto per PostgreSQL come motore di database di destinazione. Aggiunge la possibilità di generare uno schema convertito come script e di salvare gli script nei file prima di applicare lo schema all'istanza database di destinazione.	14 gennaio 2016
1.0.103	Versione di manutenzione	Aggiunge la capacità di progetto offline, la possibilità di verificare se ci sono nuove versioni disponibili e la gestione della memoria e delle prestazioni.	2 dicembre 2015
1.0.101	Versione di manutenzione	Aggiunge la procedura guidata Create New Database Migration Project (Crea un nuovo progetto di migrazione del database) . Aggiunge la possibilità di salvare il report di valutazione della migrazione del database come file PDF.	19 Ottobre 2015
1.0.100	Versione di anteprima	Fornisce la guida per l'utente per la versione di anteprima di AWS Schema Conversion Tool.	7 Ottobre 2015

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.