



Panduan Pengguna

AWS Schema Conversion Tool



Versi 1.0.672

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

AWS Schema Conversion Tool: Panduan Pengguna

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

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

Table of Contents

Apa itu AWS SCT	1
Ikhtisar konversi skema	5
Memberikan umpan balik	7
Instalasi dan Konfigurasi AWS SCT	8
Instalasi AWS SCT	8
Memvalidasi instalasi	10
Memverifikasi checksum file AWS SCT	10
Memverifikasi AWS SCT RPM file di Fedora	11
Memverifikasi AWS SCT DEB file di Ubuntu	11
Memverifikasi AWS SCT MSI file di Microsoft Windows	12
Menginstal JDBC driver	12
Menginstal JDBC driver di Linux	16
Menyimpan jalur driver di pengaturan global	17
Memperbarui AWS SCT	18
AWS Schema Conversion Tool CLI	19
AWS SCT antarmuka pengguna	20
Jendela proyek	20
Memulai dan Mengelola proyek	22
Menggunakan Wizard	23
Menyimpan proyek	26
Menambahkan server basis data	27
Mode offline	28
Filter pohon	29
.....	30
Mengimpor daftar file untuk filter pohon	32
Menyembunyikan skema	33
Laporan penilaian	34
Mengkonversi skema	38
Menerapkan skema yang dikonversi	41
Mengelola profil	42
Menyimpan AWS kredensi	42
Menyetel profil default untuk proyek	44
Izin untuk menggunakan profil AWS layanan	45
Mengkonfigurasi Secrets Manager	46

Menyimpan kata sandi basis data	47
UNIONALLmelihat	47
Pintasan keyboard	48
Memulai	50
Menghubungkan ke database sumber	52
Menghubungkan ke Amazon RDS dan dan Aurora yang dienkripsi	53
Menghubungkan ke Apache Cassandra	56
Menghubungkan ke Apache Cassandra sebagai sumber	56
Menghubungkan ke Apache Hadoop	58
Prasyarat untuk menggunakan Apache Hadoop sebagai sumber	59
Izin untuk Hive sebagai sumber	60
Izin untuk HDFS sebagai sumber	60
Izin untuk HDFS sebagai target	61
Menghubungkan ke Apache Hadoop sebagai sumber	61
Menghubungkan ke Hive dan HDFS	63
Menghubungkan ke Amazon EMR sebagai target	66
Menghubungkan ke Apache Oozie	69
Prasyarat	69
Menghubungkan ke Apache Oozie sebagai sumber	70
Izin untuk AWS Lambda	71
Menghubungkan AWS Step Functions sebagai target	73
Menghubungkan ke Azure SQL	75
Hak istimewa untuk Database Azure SQL	75
Menghubungkan ke Azure SQL Database sebagai sumber	75
Menghubungkan ke IBM DB2 untuk z/OS	77
Prasyarat untuk Db2 untuk z/OS	77
Hak istimewa untuk Db2 untuk z/OS	78
Menghubungkan ke Db2 untuk z/OS sebagai sumber	79
Hak istimewa untuk saya SQL sebagai target	81
Hak istimewa untuk Postgre SQL sebagai target	83
Db2 untuk z/OS ke pengaturan konversi Postgre SQL	84
IBMDatabase Db2 LUW	85
Hak istimewa untuk Db2 LUW	86
Menghubungkan ke Db2 LUW sebagai sumber	88
Db2 ke Postgre LUW SQL	90
Db2 ke My LUW SQL	93

Menggunakan My SQL sebagai sumber	94
Keistimewaan untuk Saya SQL	95
Menghubungkan ke My SQL sebagai sumber	95
Hak istimewa untuk Postgre SQL sebagai target	98
Database Oracle	98
Keistimewaan untuk Oracle	99
Menghubungkan ke Oracle sebagai sumber	100
Oracle ke Postgre SQL	103
Oracle ke My SQL	110
Oracle ke Amazon RDS untuk Oracle	120
Database Postgre SQL	127
Hak Istimewa untuk Postgre SQL	127
Menghubungkan ke Postgre SQL sebagai sumber	128
Keistimewaan untuk Saya SQL sebagai target	130
SAPbasis data	131
Keistimewaan untuk SAP ASE	132
Menghubungkan ke SAP ASE sebagai sumber	133
Keistimewaan untuk Saya SQL sebagai target	135
SAPASE ke pengaturan SQL konversi saya	137
Hak istimewa untuk Postgre SQL sebagai target	137
SAPASE ke pengaturan konversi Postgre SQL	138
SQLDatabase server	139
Hak Istimewa untuk Microsoft Server SQL	140
Menggunakan Otentikasi Windows dengan Microsoft Server SQL	141
Menghubungkan ke SQL Server sebagai sumber	143
SQLServer ke My SQL	146
SQLServer ke Postgre SQL	151
SQLServer ke RDS SQL Server Amazon	187
Gudang data	188
Amazon Redshift	189
Azure Synapse Analytics sebagai sumber	195
BigQuery sebagai sumber	200
Database Greenplum	206
Database Netezza	212
Gudang data Oracle	222
Kepingan salju	229

SQLGudang Data Server	238
Database Teradata	244
Database Vertica	260
Pemetaan tipe data	267
Pemetaan tipe data baru	268
Mengedit pemetaan tipe data	268
Pemetaan target virtual	269
Keterbatasan pemetaan tipe data	270
Laporan	272
Laporan penilaian	272
Buat laporan penilaian	273
Melihat laporan penilaian	274
Menyimpan laporan penilaian	278
Mengkonfigurasi laporan penilaian	280
Laporan penilaian multiserver	284
Mengkonversi skema	293
Menerapkan aturan migrasi	295
Membuat aturan migrasi	296
Mengekspor aturan migrasi	298
Mengonversi skema secara manual	298
Mengubah skema	298
Mengedit skema yang dikonversi	301
Membersihkan skema yang dikonversi	302
Mengkonversi skema secara manual	302
Memodifikasi skema sumber Anda	302
Memodifikasi skema target Anda	302
Memperbarui dan menyegarkan skema	303
Menyimpan dan menerapkan skema yang dikonversi	304
Menyimpan skema konversi Anda	304
Menerapkan skema konversi Anda	305
Skema paket ekstensi	305
Membandingkan skema	306
Melihat objek yang ditransformasikan terkait	308
Mengonversi skema gudang data	309
Izin untuk Amazon Redshift	310
Memilih strategi dan aturan optimasi	312

Mengumpulkan atau mengunggah statistik	313
Membuat aturan migrasi	315
Membuat aturan migrasi	315
Mengekspor aturan migrasi	317
Mengonversi skema Anda	318
Mengubah skema	318
Mengedit skema yang dikonversi	321
Membersihkan skema yang dikonversi	322
Mengelola dan menyesuaikan kunci	322
Topik terkait	323
Membuat dan menggunakan laporan penilaian	323
Membuat laporan penilaian migrasi database	323
Ringkasan	324
Item tindakan	326
Menyimpan laporan penilaian	326
Menangani konversi manual	327
Memodifikasi skema sumber Anda	328
Memodifikasi skema target Anda	328
Memperbarui dan menyegarkan skema konversi Anda	329
Menyimpan dan menerapkan skema konversi Anda	329
Menyimpan skema konversi Anda ke file	330
Menerapkan skema konversi Anda	330
Skema paket ekstensi	331
Pustaka Python	332
Mengonversi data Amazon Redshift	332
Mengoptimalkan database Amazon Redshift Anda	332
Mengonversi Data Menggunakan ETL	334
ETLproses	335
Prasyarat	336
AWS Glue Katalog Data	337
Batasan	337
Langkah 1: Buat proyek baru	339
Langkah 2: Buat AWS Glue pekerjaan	340
ETLproses menggunakan Python	341
Langkah 1: Buat database	342
Langkah 2: Buat koneksi	342

Langkah 3: Buat AWS Glue crawler	344
Skrip informatika ETL	346
SSISpaket	351
SSISKomponen yang didukung	355
SSISuntuk AWS Glue Studio	357
Prasyarat	357
Menambahkan SSIS paket ke AWS SCT proyek Anda	359
Mengonversi paket SSIS	360
Menciptakan AWS Glue Studio pekerjaan	361
Membuat laporan penilaian SSIS konversi	363
SSISKomponen yang didukung	363
Skrip Teradata BTEQ	364
Menambahkan BTEQ skrip ke proyek Anda AWS SCT	366
Mengkonfigurasi variabel substitusi dalam skrip BTEQ	366
Mengonversi skrip BTEQ	367
Mengelola BTEQ skrip	368
Membuat laporan penilaian konversi BTEQ skrip	369
Mengedit dan menyimpan BTEQ skrip Anda yang telah dikonversi	370
skrip shell	370
Menambahkan skrip shell ke proyek Anda AWS SCT	371
Mengkonfigurasi variabel substitusi dalam skrip shell	372
Mengonversi skrip shell	373
Mengelola skrip shell	373
Membuat laporan penilaian konversi skrip shell	374
Mengedit dan menyimpan skrip shell Anda yang telah dikonversi	375
FastExport skrip	375
Menambahkan skrip FastExport pekerjaan ke proyek Anda AWS SCT	376
Mengkonfigurasi variabel substitusi dalam skrip pekerjaan FastExport	377
Mengonversi skrip FastExport pekerjaan	378
Mengelola skrip FastExport pekerjaan	379
Membuat laporan penilaian konversi skrip FastExport pekerjaan	379
Mengedit dan menyimpan skrip FastExport pekerjaan Anda yang telah dikonversi	380
FastLoad skrip	381
Menambahkan skrip FastLoad pekerjaan ke proyek Anda AWS SCT	381
Mengkonfigurasi variabel substitusi dalam skrip pekerjaan FastLoad	382
Mengonversi skrip FastLoad pekerjaan	384

Mengelola skrip FastLoad pekerjaan	385
Membuat laporan penilaian konversi skrip FastLoad pekerjaan	386
Mengedit dan menyimpan skrip FastLoad pekerjaan Anda yang telah dikonversi	387
MultiLoad skrip	387
Menambahkan skrip MultiLoad pekerjaan ke proyek Anda AWS SCT	388
Mengonfigurasi variabel substitusi dalam skrip pekerjaan MultiLoad	389
Mengonversi skrip MultiLoad pekerjaan	390
Mengelola skrip MultiLoad pekerjaan	391
Membuat laporan penilaian konversi skrip MultiLoad pekerjaan	391
Mengedit dan menyimpan skrip MultiLoad pekerjaan Anda yang telah dikonversi	392
Migrasi kerangka kerja data besar	393
Memigrasi beban kerja Hadoop	393
Gambaran Umum	394
Langkah 1: Connect ke cluster Hadoop Anda	395
Langkah 2: Siapkan aturan pemetaan	395
Langkah 3: Buat laporan penilaian	397
Langkah 4: Migrasikan cluster Apache Hadoop Anda ke Amazon EMR	398
Menjalankan CLI skrip Anda	399
Mengelola proyek migrasi	400
Mengonversi alur kerja Oozie;	402
Gambaran Umum	402
Langkah 1: Connect ke layanan sumber dan target Anda	404
Langkah 2: Siapkan aturan pemetaan	404
Langkah 3: Konfigurasi parameter	405
Langkah 4: Buat laporan penilaian	407
Langkah 5: Konversi alur kerja Apache Oozie Anda ke AWS Step Functions	408
Menjalankan CLI skrip Anda	411
Node yang didukung	411
Integrasi dengan AWS DMS	413
Menggunakan agen AWS SCT replikasi dengan AWS DMS	413
Menggunakan agen ekstraksi AWS SCT data dengan AWS DMS	413
Meningkatkan level logging saat menggunakan AWS SCT dengan AWS DMS	413
Migrasi dari gudang data	415
Prasyarat	417
Pengaturan Amazon S3	418
Dengan asumsi peran IAM	419

Pengaturan keamanan	420
Pengaturan konfigurasi	421
Instalasi agen	421
Mengkonfigurasi agen	423
Menginstal dan mengonfigurasi agen penyalinan khusus	425
Agen awal	427
Mendaftarkan agen	428
Menyembunyikan dan memulihkan informasi untuk agen AWS SCT	428
Membuat aturan migrasi data	430
Mengubah pengaturan ekstraktor dan salin untuk migrasi data	431
Menyortir data	434
Membuat, menjalankan, dan memantau AWS SCT tugas	436
Mengekspor dan mengimpor tugas ekstraksi data	439
Ekstraksi data menggunakan perangkat AWS Snowball Edge	440
tep-by-step Prosedur S untuk memigrasi data menggunakan AWS SCT dan AWS Snowball Edge	441
Output tugas ekstraksi data	444
Menggunakan partisi virtual	446
Batas saat membuat partisi virtual	446
RANGEjenis partisi	447
LISTjenis partisi	448
DATEAUTOSPLITjenis partisi	449
Menggunakan partisi asli	450
Bekerja dengan LOBs	451
Praktik terbaik dan pemecahan masalah	452
Mengonversi aplikasi SQL	454
Ikhtisar aplikasi konversi SQL	454
SQLkode	455
Membuat proyek konversi aplikasi generik	455
Mengelola proyek konversi aplikasi	459
Menganalisis dan mengonversi kode Anda SQL	460
Membuat dan menggunakan laporan penilaian	461
Mengedit dan menyimpan SQL kode konversi Anda	462
SQLkode dalam aplikasi C #	463
Membuat proyek konversi aplikasi C #	463
Mengonversi kode aplikasi C# Anda SQL	465

Menyimpan kode aplikasi Anda yang telah dikonversi	466
Mengelola proyek konversi aplikasi C #	466
Membuat laporan penilaian konversi aplikasi C#	467
SQLkode dalam C ++	468
Membuat proyek konversi aplikasi C ++	469
Mengonversi kode aplikasi C++ Anda SQL	470
Menyimpan kode aplikasi Anda yang telah dikonversi	472
Mengelola proyek konversi aplikasi C ++	472
Membuat laporan penilaian konversi aplikasi C++	474
SQLkode di Java	475
Membuat proyek konversi aplikasi Java	475
Mengonversi kode aplikasi SQL Java Anda	478
Menyimpan kode aplikasi Anda yang telah dikonversi	480
Mengelola proyek konversi aplikasi Java	480
Membuat laporan penilaian konversi aplikasi Java	481
SQLkode dalam Pro*C	483
Membuat proyek konversi aplikasi Pro* C	483
Mengonversi kode aplikasi Pro* C Anda SQL	484
Mengedit dan menyimpan kode aplikasi Anda yang telah dikonversi	486
Mengelola proyek konversi aplikasi Pro* C	487
Membuat laporan penilaian konversi aplikasi Pro*C	487
Paket ekstensi	490
Izin untuk menggunakan paket ekstensi	491
Menggunakan skema paket ekstensi	493
Pustaka khusus untuk paket ekstensi	493
Menerapkan paket ekstensi	494
Menggunakan fungsi Lambda dari paket ekstensi AWS SCT	496
Menggunakan AWS Lambda fungsi untuk meniru fungsionalitas database	496
Menerapkan paket ekstensi untuk mendukung fungsi Lambda	496
Mengkonfigurasi fungsi paket ekstensi	498
Praktik terbaik	500
Mengkonfigurasi memori tambahan	500
Folder proyek default	500
Meningkatkan kecepatan migrasi data	501
Meningkatkan informasi pencatatan	501
Pemecahan Masalah	504

Tidak dapat memuat objek dari database sumber Oracle	504
Pesan peringatan	504
CLISReferensi	506
Prasyarat	506
Mode interaktif	506
Contoh	508
Mendapatkan CLI skenario	508
Contoh	512
CLISkenario pengeditan	512
Mode skrip	514
Contoh	514
Bahan referensi	515
Catatan rilis	516
Catatan rilis — 676	516
Catatan rilis — 675	521
Catatan rilis — 674	523
Catatan rilis — 673	531
Catatan rilis — 672	536
Catatan rilis — 671	544
Catatan rilis — 670	553
Catatan rilis — 669	558
Catatan rilis — 668	563
Catatan rilis — 667	569
Catatan rilis — 666	574
Catatan rilis — 665	578
Catatan rilis — 664	581
Catatan rilis — 663	585
Catatan rilis — 662	588
Catatan rilis — 661	593
Catatan rilis — 660	597
Catatan rilis — 659	601
Catatan rilis — 658	606
Catatan rilis — 657	611
Catatan rilis — 656	615
Catatan rilis — 655	618
Catatan rilis — 654	622

Catatan rilis — 653	625
Catatan rilis — 652	627
Catatan rilis — 651	630
Catatan rilis — 650	632
Catatan rilis — 649	634
Catatan rilis — 648	636
Catatan rilis — 647	638
Catatan rilis — 646	640
Catatan rilis — 645	641
Catatan rilis — 644	643
Catatan rilis — 642	645
Catatan rilis — 641	646
Catatan rilis — 640	647
Rilis 1.0.640 perubahan Oracle	648
Rilis 1.0.640 perubahan Microsoft Server SQL	654
Rilis 1.0.640 Perubahan Saya SQL	658
Rilis 1.0.640 Perubahan Postgre SQL	659
Rilis 1.0.640 perubahan Db2 LUW	662
Rilis 1.0.640 Perubahan Teradata	663
Rilis 1.0.640 perubahan untuk mesin lain	665
Riwayat dokumen	667
Pembaruan sebelumnya	683
.....	dcxci

Apa itu AWS Schema Conversion Tool?

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skema database yang ada dari satu mesin database ke mesin database lainnya. Anda dapat mengonversi OLTP skema relasional, atau skema gudang data. Skema konversi Anda cocok untuk Amazon Relational Database Service (RDSAmazon) MySQL, MariaDB, OracleSQL, Server, Postgre DB, Amazon Aurora DB clusterSQL, atau Amazon Redshift cluster. Skema yang dikonversi juga dapat digunakan dengan database pada EC2 instans Amazon atau disimpan sebagai data di bucket Amazon S3.

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

AWS SCT mendukung OLTP konversi berikut.

Basis data sumber	Basis data target
IBMDB2 untuk z/OS (versi 12)	Amazon Aurora My SQL -Compatible Edition (Aurora My), SQL Amazon Aurora Postgre SQL -Edisi Kompatibel (Aurora Postgre), My, Postgre SQL SQL SQL Untuk informasi selengkapnya, lihat Menghubungkan ke IBM DB2 untuk z/OS .
IBMDB2 LUW (versi 9.1, 9.5, 9.7, 10.5, 11.1, dan 11.5)	Aurora Saya, Aurora Postgre, SQL MariaDB, Saya, SQL Postgre SQL SQL Untuk informasi selengkapnya, lihat IBMDataba se Db2 LUW .
Database Microsoft Azure SQL	Aurora Saya, Aurora SQL Postgre, Saya, Postgre SQL SQL SQL

Basis data sumber	Basis data target
	<p>Untuk informasi selengkapnya, lihat Menghubungkan ke Azure SQL.</p>
<p>Microsoft SQL Server (versi 2008 R2, 2012, 2014, 2016, 2017, 2019, dan 2022)</p>	<p>Aurora My, SQL Aurora PostgreSQL, Babelfish untuk Aurora Postgre (SQLhanya untuk laporan penilaian), MariaDB, Microsoft Server, My, Postgre SQL SQL SQL</p> <p>Untuk informasi selengkapnya, lihat SQLDatabase server.</p>
<p>Saya SQL (versi 5.5 dan lebih tinggi)</p>	<p>Aurora Postgre, Saya, Postgre SQL SQL SQL</p> <p>Untuk informasi selengkapnya, lihat Menggunakan My SQL sebagai sumber.</p> <p>Anda dapat memigrasikan skema dan data dari My SQL ke cluster Aurora My SQL DB tanpa menggunakan AWS SCT Untuk informasi selengkapnya, lihat Memigrasi data ke klaster DB Amazon Aurora.</p>
<p>Oracle (versi 10.1 dan lebih tinggi)</p>	<p>Aurora Saya, Aurora Postgre, SQL MariaDB, Saya, SQL Oracle, Postgre SQL SQL</p> <p>Untuk informasi selengkapnya, lihat Database Oracle.</p>
<p>Postgre SQL (versi 9.1 dan lebih tinggi)</p>	<p>Aurora Saya, Aurora SQL Postgre, Saya, Postgre SQL SQL SQL</p> <p>Untuk informasi selengkapnya, lihat Database Postgre SQL.</p>

Basis data sumber	Basis data target
SAPASE(versi 12.5.4, 15.0.2, 15.5, 15.7, dan 16.0)	<p>Aurora Saya, Aurora Postgre, SQL MariaDB, Saya, SQL Postgre SQL SQL</p> <p>Untuk informasi selengkapnya, lihat SAPbasis data.</p>

AWS SCT mendukung konversi gudang data berikut.

Gudang data sumber	Gudang data target
Amazon Redshift	<p>Amazon Redshift</p> <p>Untuk informasi selengkapnya, lihat Amazon Redshift.</p>
Analisis Sinaps Azure	<p>Amazon Redshift</p> <p>Untuk informasi selengkapnya, lihat Azure Synapse Analytics sebagai sumber.</p>
BigQuery	<p>Amazon Redshift</p> <p>Untuk informasi selengkapnya, lihat BigQuery sebagai sumber.</p>
Database Greenplum (versi 4.3 dan 6.21)	<p>Amazon Redshift</p> <p>Untuk informasi selengkapnya, lihat Database Greenplum.</p>
Microsoft SQL Server (versi 2008 dan lebih tinggi)	<p>Amazon Redshift</p> <p>Untuk informasi selengkapnya, lihat SQLGudang Data Server.</p>
Netezza (versi 7.0.3 dan lebih tinggi)	<p>Amazon Redshift</p>

Gudang data sumber	Gudang data target
	Untuk informasi selengkapnya, lihat Database Netezza .
Oracle (versi 10.1 dan lebih tinggi)	Amazon Redshift Untuk informasi selengkapnya, lihat Gudang data Oracle .
Kepingan salju (versi 3)	Amazon Redshift Untuk informasi selengkapnya, lihat Kepingan salju .
Teradata (versi 13 dan lebih tinggi)	Amazon Redshift Untuk informasi selengkapnya, lihat Database Teradata .
Vertica (versi 7.2.2 dan lebih tinggi)	Amazon Redshift Untuk informasi selengkapnya, lihat Database Vertica .

AWS SCT mendukung data berikut Tidak ada konversi SQL database.

Basis data sumber	Basis data target
Apache Cassandra (versi 2.1.x, 2.2.16, dan 3.11.x)	Amazon DynamoDB Untuk informasi selengkapnya, lihat Menghubungkan ke Apache Cassandra .

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

Sumber	Target
Skrip informatika ETL	Informatika
ETLPaket Layanan Integrasi Microsoft SQL Server (SSIS)	AWS Glue atau AWS Glue Studio
Skrip shell dengan perintah tertanam dari Teradata Basic Teradata Query () BTEQ	Pergeseran Merah Amazon RSQL
Skrip Teradata BTEQ ETL	AWS Glue atau Amazon Redshift RSQL
Skrip pekerjaan Teradata FastExport	Pergeseran Merah Amazon RSQL
Skrip pekerjaan Teradata FastLoad	Pergeseran Merah Amazon RSQL
Skrip pekerjaan Teradata MultiLoad	Pergeseran Merah Amazon RSQL

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

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

Ikhtisar konversi skema

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

Untuk informasi tentang cara menginstal AWS SCT, lihat [Instalasi dan Konfigurasi AWS Schema Conversion Tool](#).

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

Untuk informasi tentang proses konversi, lihat [Mengonversi skema database di AWS Schema Conversion Tool](#).

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

- Anda dapat menggunakan agen ekstraksi data untuk mengekstrak data dari gudang data Anda guna mempersiapkan migrasi ke Amazon Redshift. Untuk mengelola agen ekstraksi data, Anda dapat menggunakan AWS SCT. Untuk informasi selengkapnya, lihat [Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool](#).
- Anda dapat menggunakan AWS SCT untuk membuat AWS DMS titik akhir dan tugas. Anda dapat menjalankan dan memantau tugas-tugas ini dari AWS SCT. Untuk informasi selengkapnya, lihat [Integrasi dengan AWS Database Migration Service AWS Schema Conversion Tool](#).
- Dalam beberapa kasus, fitur database tidak dapat dikonversi ke fitur Amazon RDS atau Amazon Redshift yang setara. Wizard paket AWS SCT ekstensi dapat membantu Anda menginstal AWS Lambda fungsi dan pustaka Python untuk meniru fitur yang tidak dapat dikonversi. Untuk informasi selengkapnya, lihat [Menggunakan paket ekstensi dengan AWS Schema Conversion Tool](#).
- Anda dapat menggunakan AWS SCT untuk mengoptimalkan database Amazon Redshift yang ada. AWS SCT merekomendasikan kunci sortir dan kunci distribusi untuk mengoptimalkan database Anda. Untuk informasi selengkapnya, lihat [Mengonversi data dari Amazon Redshift menggunakan AWS Schema Conversion Tool](#).
- Anda dapat menggunakan AWS SCT untuk menyalin skema database lokal yang ada ke instans Amazon RDS DB yang menjalankan mesin yang sama. Anda dapat menggunakan fitur ini untuk menganalisis potensi penghematan biaya pindah ke cloud dan mengubah jenis lisensi Anda.
- Anda dapat menggunakan AWS SCT untuk mengkonversi SQL dalam C ++, C #, Java, atau kode aplikasi lainnya. Anda dapat melihat, menganalisis, mengedit, dan menyimpan SQL kode yang dikonversi. Untuk informasi selengkapnya, lihat [Mengonversi aplikasi menggunakan SQL AWS SCT](#).
- Anda dapat menggunakan AWS SCT untuk memigrasikan proses ekstraksi, transformasi, dan load (ETL). Untuk informasi selengkapnya, lihat [Mengonversi Data Menggunakan ETL Proses di AWS Schema Conversion Tool](#).

Memberikan umpan balik

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

Untuk memberikan umpan balik tentang AWS SCT

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

Instalasi dan Konfigurasi AWS Schema Conversion Tool

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

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

AWS SCT tersedia sebagai aplikasi mandiri dan alat baris perintah. Untuk informasi tentang alat baris perintah, lihat [AWS Schema Conversion Tool CLI](#).

Topik

- [Instalasi AWS Schema Conversion Tool](#)
- [Memvalidasi installation AWS Schema Conversion Tool](#)
- [Menginstal JDBC driver untuk AWS Schema Conversion Tool](#)
- [Memperbarui AWS Schema Conversion Tool](#)
- [AWS Schema Conversion Tool CLI](#)

Instalasi AWS Schema Conversion Tool

Anda dapat menginstal AWS SCT pada sistem operasi berikut:

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

Untuk menginstal AWS SCT

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

- [Microsoft Windows](#)

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

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

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

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

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

4. Unduh driver Java Database Connectivity (JDBC) untuk mesin basis data sumber dan target Anda. Untuk petunjuk dan tautan unduhan, lihat [Menginstal JDBC driver untuk AWS Schema Conversion Tool](#).

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

Memvalidasi installation AWS Schema Conversion Tool

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

Bagian ini mencakup topik berikut.

Topik

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

Memverifikasi checksum file AWS SCT

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

Untuk memverifikasi file AWS SCT distribusi menggunakan checksum

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

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

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

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

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

Memverifikasi AWS SCT RPM file di Fedora

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

Untuk memverifikasi AWS SCT RPM file di Fedora

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

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

6. Periksa apakah impor berhasil dengan menjalankan perintah berikut:

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

7. Periksa RPM tanda tangan dengan menjalankan perintah berikut:

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

Memverifikasi AWS SCT DEB file di Ubuntu

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

Untuk memverifikasi AWS SCT DEB file di Ubuntu

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

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

7. Verifikasi tanda tangan dengan menjalankan perintah berikut:

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

Memverifikasi AWS SCT MSI file di Microsoft Windows

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

Untuk memverifikasi AWS SCT MSI file di Windows

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

Menginstal JDBC driver untuk AWS Schema Conversion Tool

AWS SCT Agar berfungsi dengan benar, unduh JDBC driver untuk mesin basis data sumber dan target Anda. Jika Anda menggunakan platform basis data target virtual, Anda tidak perlu mengunduh

JDBC driver untuk mesin basis data target Anda. Untuk informasi selengkapnya, lihat [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).

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

Anda dapat mengunduh driver database dari lokasi berikut.

⚠ Important

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

Mesin basis data	Pengemudi	Lokasi unduhan
Amazon Aurora Edisi Kompatibel Saya SQL	mysql-connector-java-5.1.6.jar	https://www.mysql.com/products/konektor/
Amazon Aurora SQL Postgre -Edisi Kompatibel	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/driver/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
Amazon Redshift Tanpa Server	redshift-jdbc42-2.1.0.9.jar	https://s3.amazonaws.com/redshift-downloads/driver/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip

Mesin basis data	Pengemudi	Lokasi unduhan
Apache Hive	hive-jdbc-2.3.4-standalone.jar	https://repo1.maven.org/maven2/org/apache/hive/hive-jdbc/2.3.4/hive-jdbc-2.3.4-standalone.jar
Database Azure SQL	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/release-notes-for-theseql/koneksi/jdbc-jdbc-driver? dilihat= sql-server-ver 15 #72
Analisis Sinaps Azure	mssql-jdbc-7.2.2.jre11.jar	https://docs.microsoft.com/en-us/release-notes-for-theseql/koneksi/jdbc-jdbc-driver? dilihat= sql-server-ver 15 #72
Database Greenplum	postgresql-42.2.19.jar	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
IBMDb2 untuk z/OS	db2jcc-db2jcc4.jar	https://www.ibm.com/support/halaman/db2-download-db2-zos jdbc-driver-versions-and
IBMDb2 LUW	db2jcc-db2jcc4.jar	https://www.ibm.com/support/halaman/node/382667
MariaDB	mariadb-java-client-2.4.1.jar	https://downloads.mariadb.com/Connectors/java/konektor-java-2.4.1/ mariadb-java-client-2.4.1.jar
SQLServer Microsoft	mssql-jdbc-10.2.jar	https://docs.microsoft.com/en-us/download-microsoft-jdbc-driversql/connect/jdbc/-? for-sql-server melihat= 15 sql-server-ver
Saya SQL	mysql-connector-java-8.0.15.jar	https://dev.mysql.com/downloads/konektor/j/

Mesin basis data	Pengemudi	Lokasi unduhan
Netezza	<p>nzjdbc.jar</p> <p>Gunakan perangkat lunak alat klien. Unduh driver versi 7.2.1, yang kompatibel dengan gudang data versi 7.2.0.</p>	<p>http://www.ibm.com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.datacon.doc/c_datacon_plg_overview.html</p>
Oracle	<p>ojdbc8.jar</p> <p>Driver versi 8 dan lebih tinggi didukung.</p>	<p>https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html</p>
Postgre SQL	<p>postgresql-42.2.19.jar</p>	<p>https://jdbc.postgresql.org/download/postgresql-42.2.19.jar</p>
SAP ASE (Sybase ASE)	<p>jconn4.jar</p>	<p>jConnect JDBC Sopir</p>
Kepingan salju	<p>snowflake-jdbc-3.9.2.jar</p> <p>Untuk informasi selengkapnya, lihat Mengunduh/ Mengintegrasikan JDBC Driver.</p>	<p>https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9.2.jar</p>
Teradata	<p>terajdbc4.jar</p> <p>tdgssconfig.jar</p> <p>Untuk JDBC driver Teradata versi 16.20.00.11 dan yang lebih tinggi, Anda tidak memerlukan file. tdgssconfig.jar</p>	<p>https://downloads.teradata.com/download/konektivitas/jdbc-driver</p>

Mesin basis data	Pengemudi	Lokasi unduhan
Vertica	vertica-jdbc-9.1.1-0.jar Versi driver 7.2.0 dan yang lebih tinggi didukung.	https://www.vertica.com/client_drivers/9.1.x/9.1.1-0/vertica-jdbc-9.1.1-0.jar

Menginstal JDBC driver di Linux

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

Untuk menginstal JDBC driver pada sistem Linux Anda

1. Buat direktori untuk menyimpan JDBC driver di.

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

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

Mesin basis data	Perintah instalasi
Amazon Aurora (Kompatibel dengan sayaSQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>
Amazon Aurora (Kompatibel dengan PostgreSQL)	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>
SQLServer Microsoft	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo tar xzvf /tmp/sqljdbc_X.X.X_enu.tar.gz</pre>
Saya SQL	<pre>PROMPT> cd /usr/local/jdbc-drivers</pre>

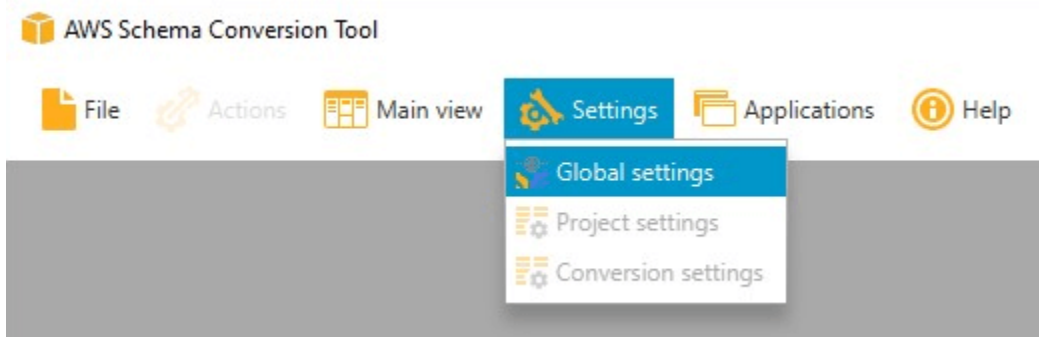
Mesin basis data	Perintah instalasi
	<pre>PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz</pre>
Oracle	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo mkdir oracle-jdbc PROMPT> cd oracle-jdbc PROMPT> sudo cp -a /tmp/ojdbc8.jar .</pre>
Postgre SQL	<pre>PROMPT> cd /usr/local/jdbc-drivers PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .</pre>

Menyimpan jalur driver di pengaturan global

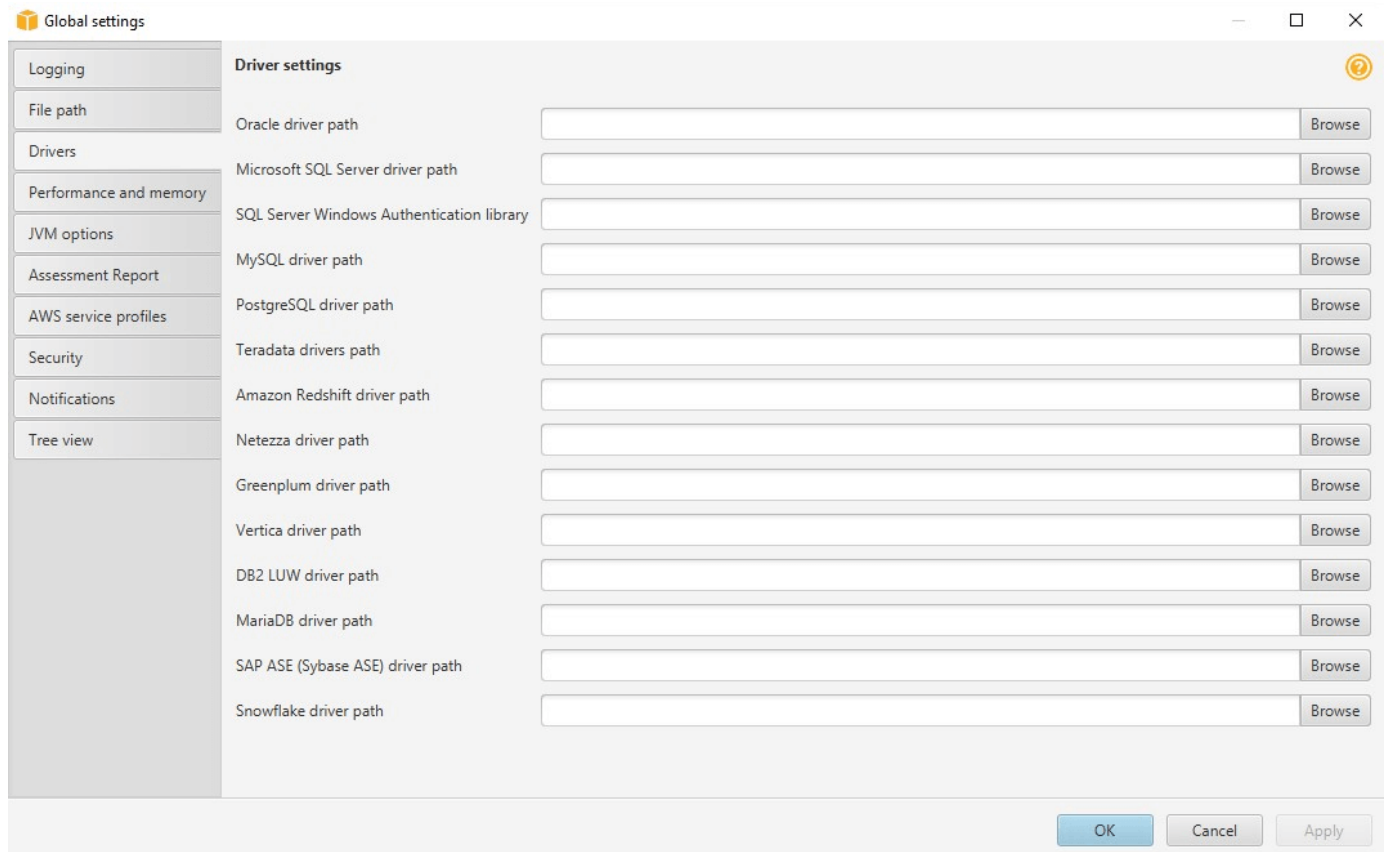
Setelah Anda mengunduh dan menginstal JDBC driver yang diperlukan, Anda dapat mengatur lokasi driver secara global dalam AWS SCT pengaturan. Jika Anda tidak mengatur lokasi driver secara global, aplikasi meminta Anda untuk lokasi driver ketika Anda terhubung ke database.

Untuk memperbarui lokasi file driver

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



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



3. Setelah Anda selesai menambahkan jalur driver, pilih OK.

Memperbarui AWS Schema Conversion Tool

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

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

Untuk memeriksa pembaruan AWS SCT

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

AWS Schema Conversion Tool CLI

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

[AWSSchemaConversionToolBatch.jar](#)

Menavigasi antarmuka pengguna AWS SCT

Gunakan topik berikut untuk membantu Anda bekerja dengan antarmuka AWS SCT pengguna. Untuk informasi tentang menginstal AWS SCT, lihat [Instalasi dan Konfigurasi AWS Schema Conversion Tool](#).

Topik

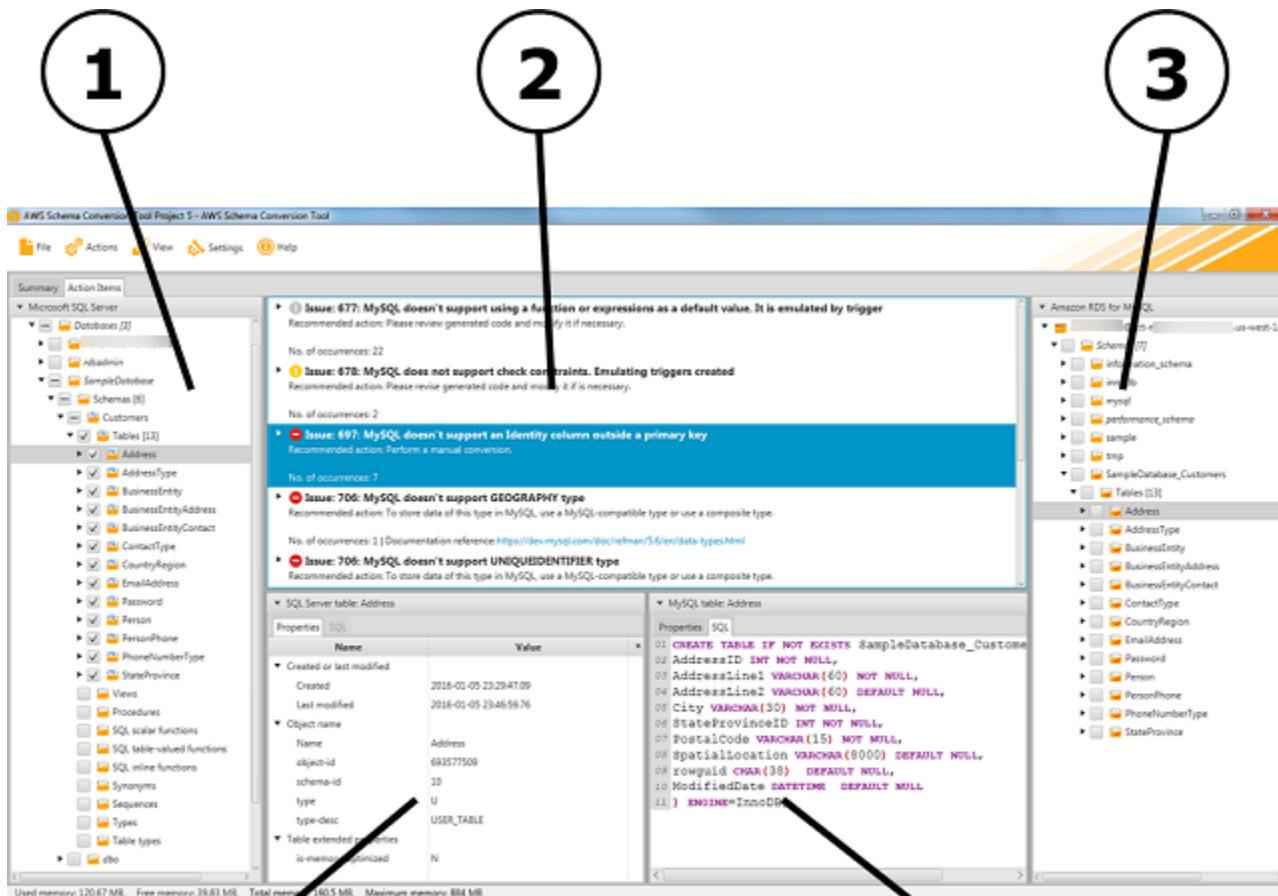
- [Melihat Jendela Proyek di AWS SCT](#)
- [Memulai dan mengelola Proyek di AWS SCT](#)
- [Mengggunakan AWS SCT Wizard](#)
- [Menyimpan proyek di AWS SCT](#)
- [Menambahkan server ke proyek di AWS SCT](#)
- [Mengggunakan mode offline di AWS Schema Conversion Tool](#)
- [Mengggunakan filter pohon di AWS Schema Conversion Tool](#)
- [Menyembunyikan skema di AWS Schema Conversion Tool](#)
- [Melihat Laporan Penilaian di AWS Schema Conversion Tool](#)
- [Mengonversi Skema di AWS Schema Conversion Tool](#)
- [Menerapkan skema yang dikonversi di AWS Schema Conversion Tool](#)
- [Mengelola Profil di AWS Schema Conversion Tool](#)
- [Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)
- [Menyimpan kata sandi di AWS Schema Conversion Tool](#)
- [Membuat UNION ALL tampilan di AWS Schema Conversion Tool](#)
- [Mengggunakan Pintasan Keyboard di AWS Schema Conversion Tool](#)

Melihat Jendela Proyek di AWS SCT

Ilustrasi berikut adalah apa yang Anda lihat AWS SCT ketika Anda membuat proyek migrasi skema, dan kemudian mengonversi skema.

1. Di panel kiri, skema dari database sumber Anda disajikan dalam tampilan pohon. Skema database Anda adalah “lazy loaded.” Dengan kata lain, ketika Anda memilih item dari tampilan pohon, AWS SCT mendapatkan dan menampilkan skema saat ini dari database sumber Anda.

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



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

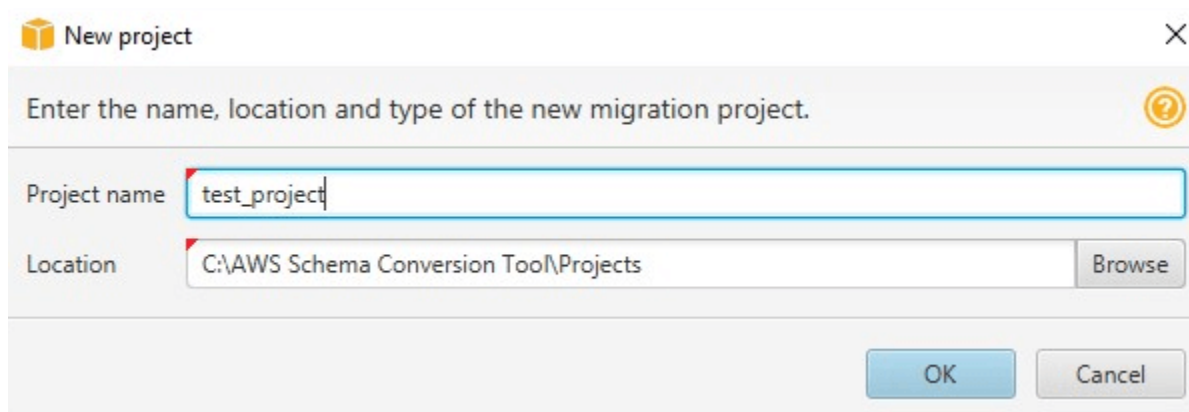
Memulai dan mengelola Proyek di AWS SCT

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

Gunakan prosedur berikut untuk membuat AWS Schema Conversion Tool proyek.

Untuk membuat proyek Anda

1. Mulai AWS Schema Conversion Tool.
2. Pada menu File, pilih Proyek baru. Kotak dialog Proyek baru muncul.



3. Memasukkan nama untuk proyek Anda, yang menyimpan secara lokal di komputer Anda.
4. Memasukkan lokasi untuk file proyek lokal Anda.
5. Pilih OK untuk membuat AWS SCT proyek Anda.
6. Pilih Tambahkan sumber untuk menambahkan database sumber baru ke AWS SCT proyek Anda. Anda dapat menambahkan beberapa database sumber ke AWS SCT proyek Anda.
7. Pilih Tambahkan target untuk menambahkan platform target baru di AWS SCT proyek Anda. Anda dapat menambahkan beberapa platform target ke AWS SCT proyek Anda.
8. Pilih skema basis data sumber di panel kiri.
9. Di panel kanan, tentukan platform basis data target untuk skema sumber yang dipilih.
10. Pilih Buat pemetaan. Tombol ini menjadi aktif setelah Anda memilih skema basis data sumber dan platform basis data target. Untuk informasi selengkapnya, lihat [Pemetaan tipe data](#).

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

Menggunakan AWS SCT Wizard

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

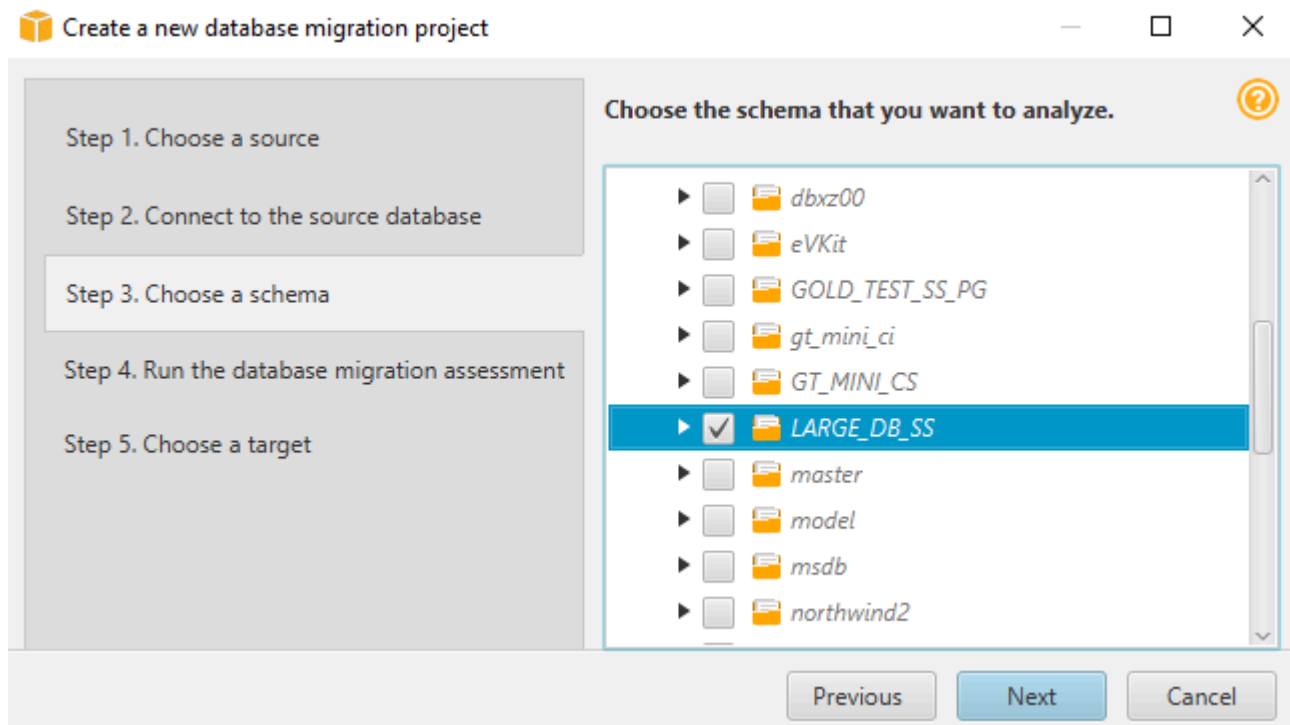
Untuk menjalankan wizard proyek baru

1. Pilih database sumber Anda.
 - a. Mulai AWS Schema Conversion Tool.
 - b. Pada menu File, pilih Wizard proyek baru. Kotak dialog Buat proyek migrasi database baru terbuka.
 - c. Untuk memasukkan informasi koneksi basis data sumber, gunakan petunjuk berikut:

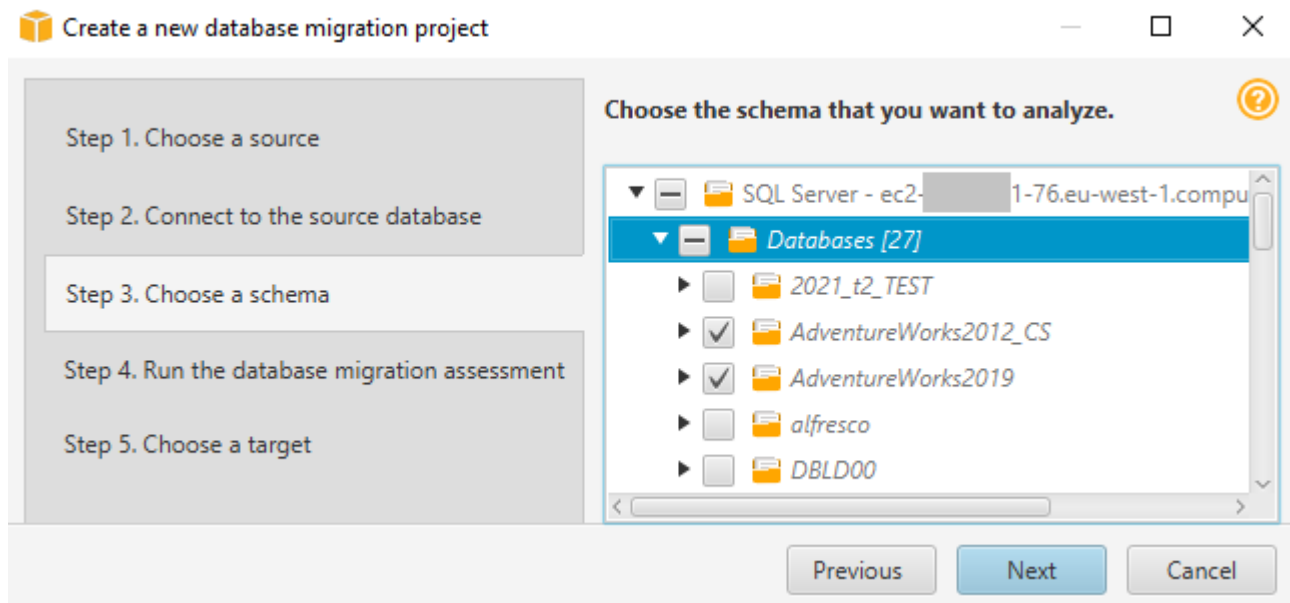
Parameter	Tindakan
Nama proyek	Memasukkan nama untuk proyek Anda, yang menyimpan secara lokal di komputer Anda.
Lokasi	Memasukkan lokasi untuk file proyek lokal Anda.
Jenis sumber	Pilih salah satu opsi berikut: SQLdatabase, Tidak ada SQL database, atau ETL. Jika Anda ingin melihat laporan ringkasan yang mencakup semua tujuan migrasi, pilih SQLdatabase.
Mesin sumber	Pilih mesin basis data sumber Anda.
Strategi migrasi	Pilih salah satu opsi berikut: <ul style="list-style-type: none">• Saya ingin mengganti mesin dan mengoptimalkan untuk cloud — Opsi ini mengubah basis data sumber Anda ke mesin database baru.

Parameter	Tindakan
	<ul style="list-style-type: none">• Saya ingin mempertahankan mesin yang sama tetapi mengoptimalkan untuk cloud — Opsi ini membuat mesin database Anda apa adanya dan memindahkan database dari lokal ke cloud.• Saya ingin melihat laporan gabungan untuk sakelar mesin database dan pengoptimalan untuk cloud — Opsi ini membandingkan kompleksitas migrasi semua opsi migrasi yang tersedia. <p>Jika Anda ingin melihat laporan penilaian gabungan yang mencakup semua tujuan migrasi, pilih opsi terakhir.</p>

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



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



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

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

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

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

Menyimpan proyek di AWS SCT

Gunakan prosedur berikut untuk menyimpan AWS Schema Conversion Tool proyek.

Untuk menyimpan proyek Anda

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

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

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

Untuk membuka proyek Anda

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

Jika Anda membuka proyek yang disimpan dalam AWS SCT versi 1.0.655 atau sebelumnya, AWS SCT secara otomatis membuat aturan pemetaan untuk semua skema basis data sumber ke platform basis data target. Untuk menambahkan platform basis data target lainnya, hapus aturan pemetaan yang ada dan kemudian buat aturan pemetaan baru. Untuk informasi selengkapnya tentang membuat aturan pemetaan, lihat [Pemetaan tipe data](#).

Menambahkan server ke proyek di AWS SCT

Anda dapat menambahkan beberapa server basis data sumber dan target ke sebuah AWS Schema Conversion Tool proyek.

Untuk menambahkan server ke proyek Anda

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

Gunakan prosedur berikut untuk terhubung ke database Anda.

Untuk terhubung ke database Anda

1. Buka menu konteks (klik kanan) untuk server database, lalu pilih Buat koneksi.

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

2. Masukkan kata sandi untuk terhubung ke server basis data sumber Anda.

3. Pilih Uji koneksi untuk memverifikasi yang AWS SCT dapat terhubung ke database sumber Anda.
4. Pilih Connect untuk terhubung ke database sumber Anda.

Gunakan prosedur berikut untuk menghapus server database dari AWS SCT proyek Anda.

Untuk menghapus server database

1. Pilih server database yang akan dihapus.
2. Buka menu konteks (klik kanan), lalu pilih Hapus dari proyek.

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

Menggunakan mode offline di AWS Schema Conversion Tool

Anda dapat menjalankan AWS Schema Conversion Tool dalam mode offline. Setelah itu, Anda dapat mempelajari cara bekerja dengan AWS SCT proyek yang ada saat terputus dari basis data sumber Anda.

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

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

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

Untuk berjalan AWS SCT dalam mode offline

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

2. Tambahkan server database sumber dan sambungkan ke database sumber Anda. Untuk informasi selengkapnya, lihat [Menambahkan server ke proyek di AWS SCT](#).
3. Tambahkan server database target atau gunakan platform basis data target virtual. Untuk informasi selengkapnya, lihat [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).
4. Buat aturan pemetaan untuk menentukan platform basis data target untuk basis data sumber Anda. Untuk informasi selengkapnya, lihat [Memetakan tipe data di AWS Schema Conversion Tool](#).
5. Pilih Tampilan, lalu pilih Tampilan utama.
6. Di panel kiri yang menampilkan objek database sumber Anda, pilih skema basis data sumber Anda. Buka menu konteks (klik kanan) untuk objek, lalu pilih Muat skema. Operasi ini memuat semua metadata skema sumber ke dalam proyek Anda. AWS SCT

Operasi Create report dan Convert schema juga memuat semua metadata skema sumber ke dalam project Anda. AWS SCT Jika Anda menjalankan salah satu operasi ini dari menu konteks, lewati operasi skema Muat.

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

Menggunakan filter pohon di AWS Schema Conversion Tool

Untuk memigrasikan data dari sumber ke target, AWS SCT memuat semua metadata dari basis data sumber dan target ke dalam struktur pohon. Struktur ini muncul AWS SCT sebagai tampilan pohon di jendela proyek utama.

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

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

Untuk filter pohon, ingatlah hal berikut:

- Filter default adalah ANY, yang berarti bahwa filter menggunakan pencarian nama untuk menemukan objek.
- Ketika Anda memilih satu atau beberapa jenis objek, Anda hanya melihat jenis objek di pohon.
- Anda dapat menggunakan filter mask untuk menampilkan berbagai jenis simbol, termasuk Unicode, spasi, dan karakter khusus. Karakter “%” adalah wildcard untuk simbol apa pun.
- Setelah Anda menerapkan filter, hitungan hanya menunjukkan jumlah objek yang difilter.

Untuk membuat filter pohon

1. Buka AWS SCT proyek yang sudah ada.
2. Connect ke database yang ingin Anda terapkan filter pohon.
3. Pilih ikon filter.



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

4. Masukkan informasi berikut di kotak dialog Filter. Opsi di kotak dialog berbeda untuk setiap mesin database.

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

AWS SCT opsi filter	Tindakan
Kondisi	<p>Untuk Kategori di Level, pilih kondisi penyaringan antara Suka dan Tidak suka.</p> <p>Untuk Status di Level, opsi kondisi penyaringan tidak tersedia.</p>
Nilai	<p>Untuk Kategori di Level, masukkan Nilai untuk memfilter pohon dengan nilai ini.</p> <p>Gunakan percent (%) sebagai wildcard untuk menampilkan semua objek.</p> <p>Untuk Status di Level, pilih Nilai antara True dan False.</p>
Dan/atau	Pilih AND atau operator OR logis untuk menerapkan beberapa klausa filter.

Filter

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

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

Add new clause

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

Import Export Download template Reset Apply Close

- Pilih Tambahkan klausa baru untuk menambahkan klausa filter tambahan. AWS SCT dapat menerapkan beberapa klausa filter menggunakan AND atau operator OR logis.

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

Saat Anda memfilter skema yang muncul di pohon, Anda tidak mengubah objek yang dikonversi saat Anda mengonversi skema Anda. Filter hanya mengubah apa yang Anda lihat di pohon.

Mengimpor daftar file untuk filter pohon

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

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

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

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

CSVfile menggunakan titik koma sebagai pemisah dan memiliki format berikut:

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

Gunakan JSON file untuk menjelaskan kasus penyaringan yang kompleks, seperti aturan bersarang. JSONfile memiliki format berikut:

- `filterGroupType` adalah jenis aturan filter (AND atau operator OR logis) yang berlaku untuk beberapa klausa filter.
- `filterCategory` adalah tingkat filter (Kategori atau Status).
- `names` adalah daftar nama objek yang berlaku untuk filter Kategori.

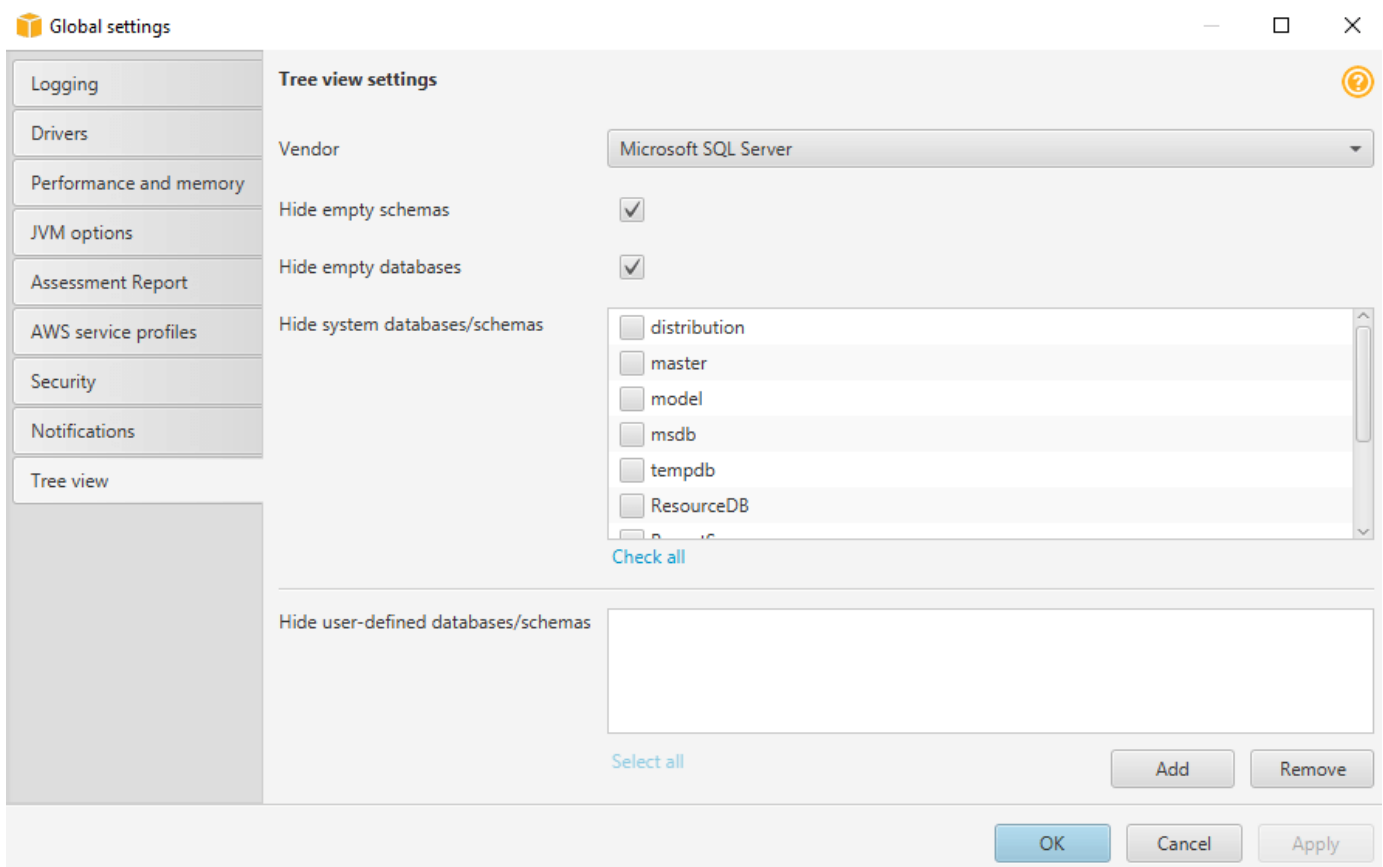
- `filterCondition` adalah kondisi penyaringan (LIKE atau NOT LIKE) yang berlaku untuk filter Kategori.
- `transformName` adalah nama status yang berlaku untuk filter Status.
- `value` adalah nilai untuk menyaring pohon dengan.
- `transformValue` adalah nilai filter (TRUE atau FALSE) yang berlaku untuk filter Status.

Menyembunyikan skema di AWS Schema Conversion Tool

Gunakan pengaturan tampilan pohon untuk menentukan skema dan database apa yang ingin Anda lihat di tampilan AWS SCT pohon. Anda dapat menyembunyikan skema kosong, database kosong, database sistem, dan database dan skema yang ditentukan pengguna.

Untuk menyembunyikan database dan skema dalam tampilan pohon

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



4. Di bagian Pengaturan tampilan pohon, lakukan hal berikut:
 - Untuk Vendor, pilih platform database.
 - Pilih Sembunyikan skema kosong untuk menyembunyikan skema kosong untuk platform database yang dipilih.
 - Pilih Sembunyikan database kosong untuk menyembunyikan database kosong untuk platform database yang dipilih.
 - Untuk Sembunyikan database/skema sistem, pilih database sistem dan skema berdasarkan nama untuk menyembunyikannya.
 - Untuk Sembunyikan database/skema yang ditentukan pengguna, masukkan nama database dan skema yang ditentukan pengguna yang ingin Anda sembunyikan, lalu pilih Tambah. Nama-nama itu tidak peka huruf besar/kecil.
5. Pilih OKE.

Melihat Laporan Penilaian di AWS Schema Conversion Tool

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

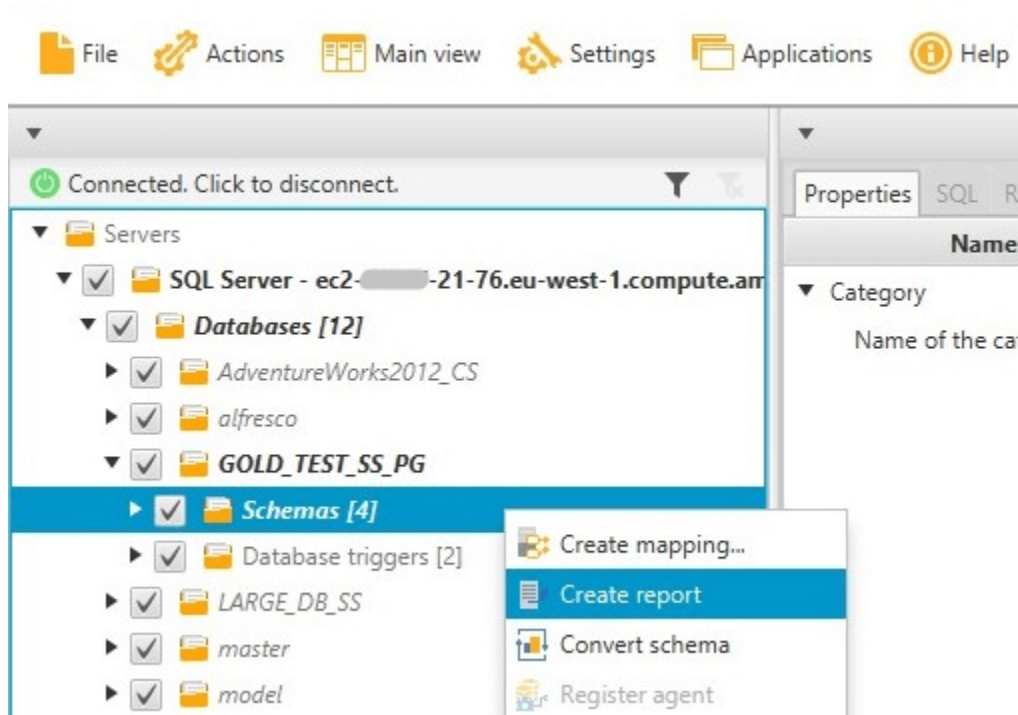
Anda dapat membuat laporan penilaian migrasi database setelah menambahkan basis data sumber dan platform target ke proyek Anda dan menentukan aturan pemetaan.

Untuk membuat dan melihat laporan penilaian migrasi database

1. Pastikan Anda membuat aturan pemetaan untuk skema database sumber untuk membuat laporan penilaian. Untuk informasi selengkapnya, lihat [Memetakan tipe data baru di AWS Schema Conversion Tool](#).
2. Pada menu Tampilan, pilih Tampilan utama.
3. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih objek skema untuk membuat laporan penilaian.

Pastikan Anda memilih kotak centang untuk semua objek skema untuk membuat laporan penilaian.

4. Buka menu konteks (klik kanan) untuk objek, lalu pilih Buat laporan.



Tampilan laporan penilaian terbuka.

5. Pilih tab Item tindakan.

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

The screenshot displays the AWS Schema Conversion Tool interface. The top navigation bar includes 'File', 'Actions', 'Assessment Report view', 'Settings', 'Applications', 'Help', 'Add source', and 'Add target'. The main window is divided into several sections:

- Summary:** Shows 'Connected. Click to disconnect.' and a tree view of the source database structure, including Servers, Databases (12), and Schemas (1).
- Action items:** A list of issues identified during conversion, such as:
 - Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required.
 - Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option.
 - Issue 794:** MySQL doesn't support user-defined data types.
 - Issue 826:** Check the default value for a DateTime variable.
 - Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values.
 - Issue 9997:** Unable to resolve objects.
 - Issue 690:** MySQL doesn't support table types.
 - Issue 811:** Unable to convert functions.
- Source Microsoft SQL Server procedure:** A detailed view of the 'POSITION_UPDATE_CASH_CGT_BULK' procedure, showing its parameters and SQL code.
- Target Amazon RDS for MySQL category:** A table showing the target schema details.

Properties	SQL	Apply status	Key management	Name	Value
Category	Name of the category			Schemas	

6. Pilih tab Ringkasan.

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

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

Contoh ringkasan laporan penilaian ditampilkan sebagai berikut.

Summary | Action Items

Save to CSV Save to PDF

Database migration assessment report

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

Executive summary

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

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

Database objects with conversion actions for Amazon RDS for PostgreSQL

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

We found 7 encrypted object(s).

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

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

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

Figure: Conversion statistics for database storage objects

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

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

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

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

Database objects with conversion actions for Amazon RDS for PostgreSQL

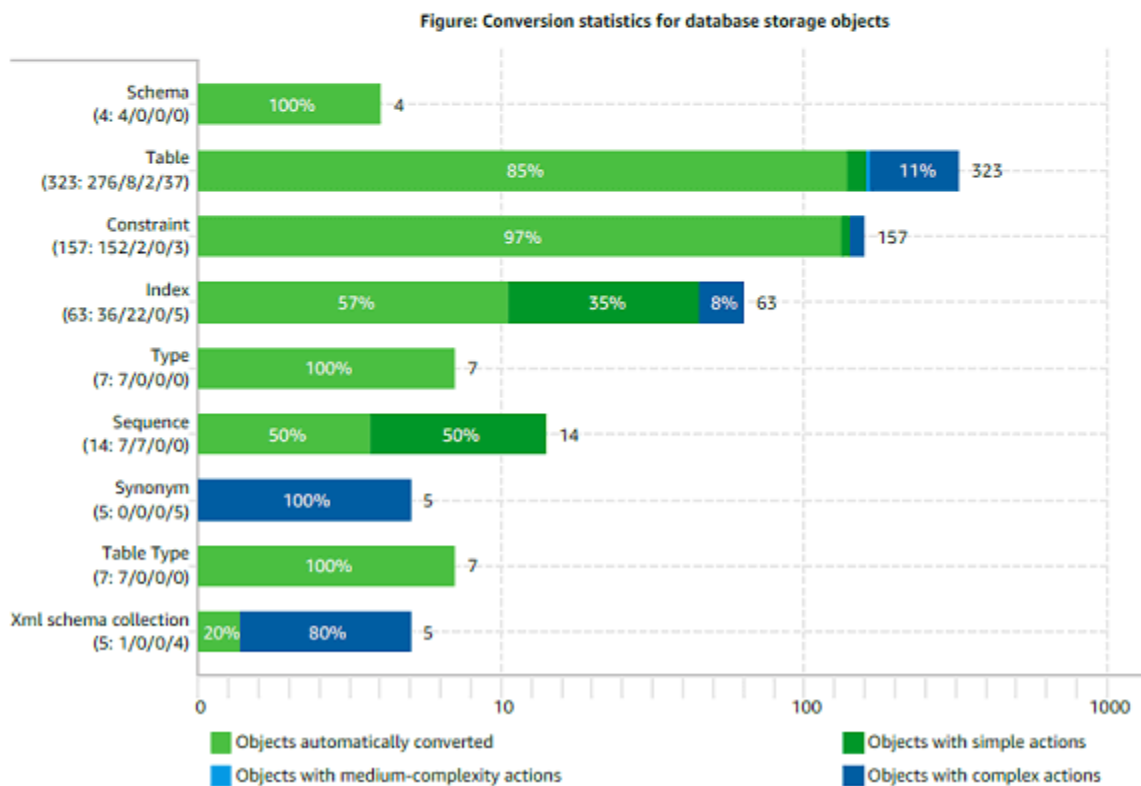
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

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

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

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

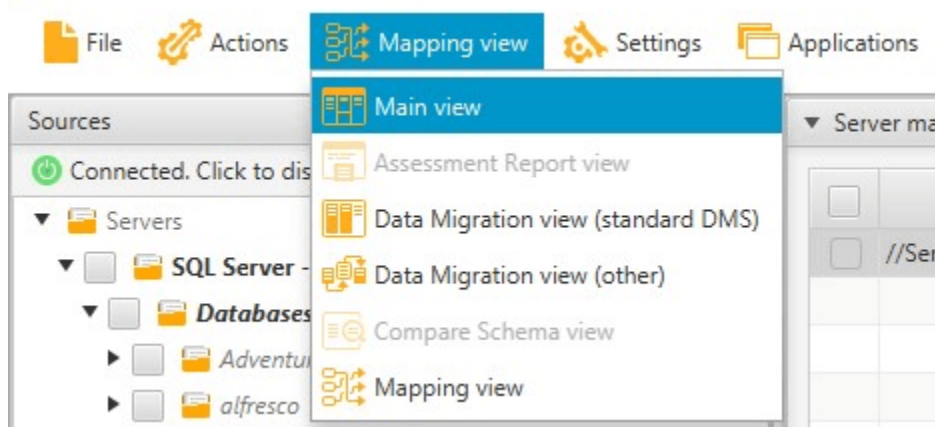


Mengonversi Skema di AWS Schema Conversion Tool

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

Untuk mengonversi skema Anda

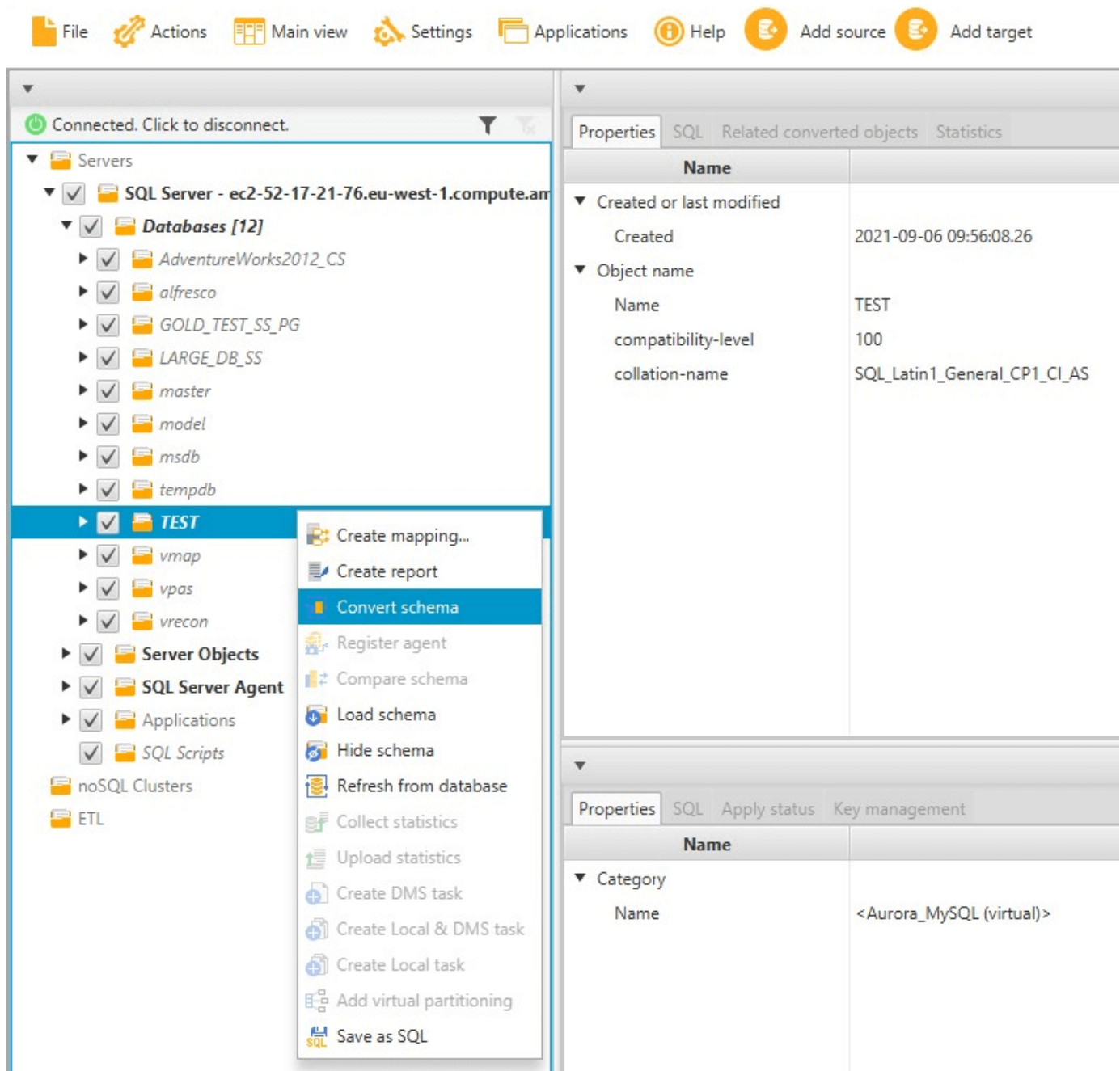
1. Pilih Tampilan, lalu pilih Tampilan utama.



2. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih kotak centang untuk nama objek yang akan dikonversi. Selanjutnya, pilih objek ini. AWS SCT menyoroti nama objek dengan warna biru. Buka menu konteks (klik kanan) untuk objek, dan pilih Konversi skema.

Untuk mengonversi beberapa objek database, pilih kotak centang untuk semua objek.

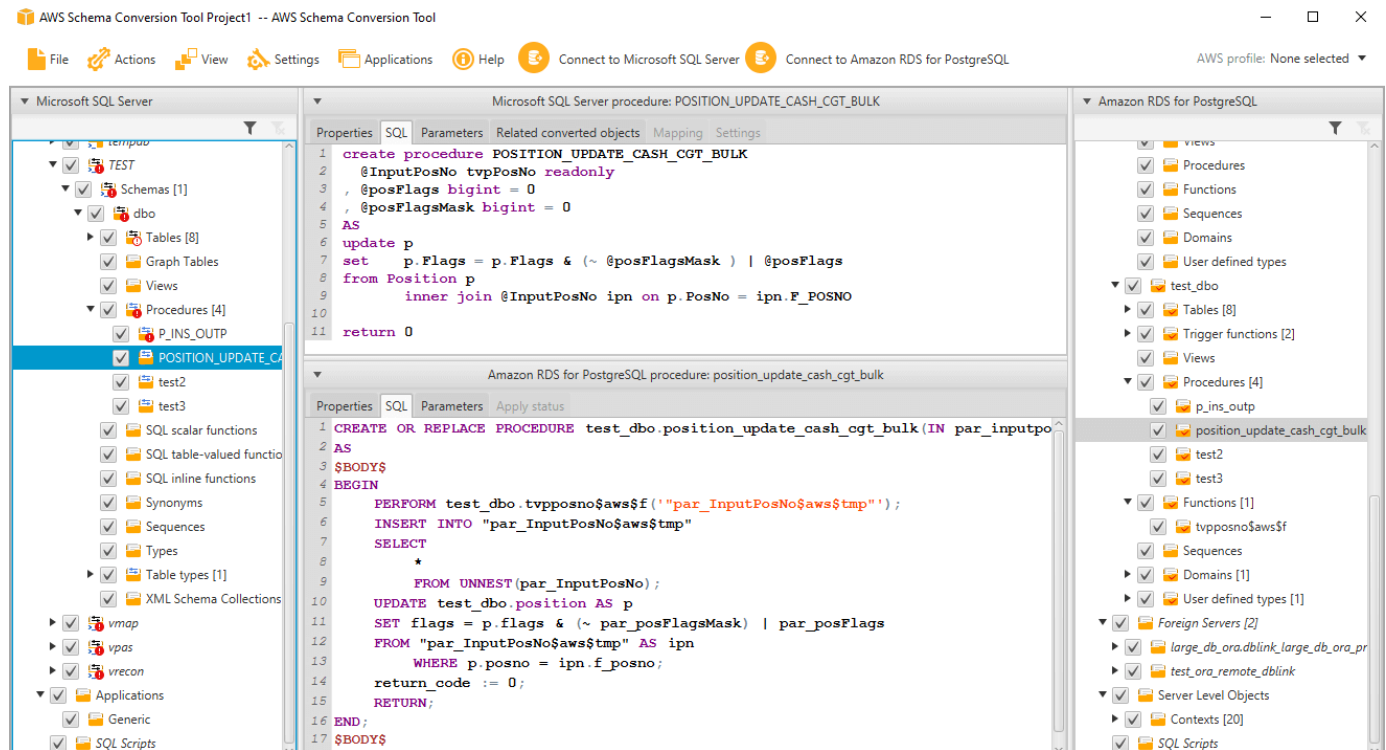
Selanjutnya, pilih node induk. Misalnya, untuk tabel, node induk adalah Tabel. Pastikan bahwa AWS SCT menyoroti nama node induk dengan warna biru. Buka menu konteks (klik kanan) untuk node induk, dan pilih Konversi skema.



- Setelah AWS SCT selesai mengonversi skema, Anda dapat melihat skema yang diusulkan di panel di sebelah kanan proyek Anda.

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

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



Menerapkan skema yang dikonversi di AWS Schema Conversion Tool

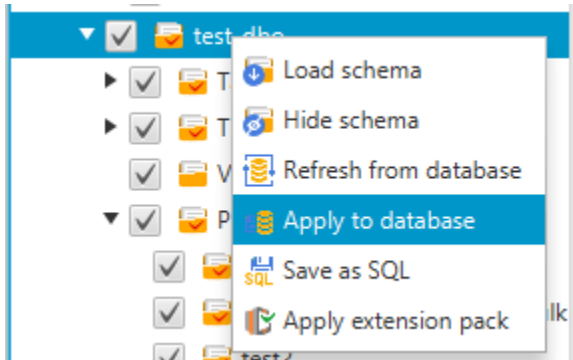
Anda dapat menerapkan skema database yang dikonversi ke instans DB target Anda. Setelah skema diterapkan ke instans DB target, Anda dapat memperbarui skema berdasarkan item tindakan dalam laporan penilaian migrasi database.

Warning

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

Untuk menerapkan skema database yang dikonversi ke instance database target Anda

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



Skema yang dikonversi diterapkan ke instans DB target.

Mengelola Profil di AWS Schema Conversion Tool

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

AWS SCT meminta Anda untuk AWS kredensial Anda ketika Anda mengakses fitur yang membutuhkannya. Anda dapat menyimpan kredensial Anda di pengaturan aplikasi global. Saat AWS SCT meminta kredensial Anda, Anda dapat memilih kredensial yang disimpan.

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

Menyimpan AWS kredensial

Gunakan prosedur berikut untuk menyimpan AWS kredensial secara global.

Untuk menyimpan AWS kredensi

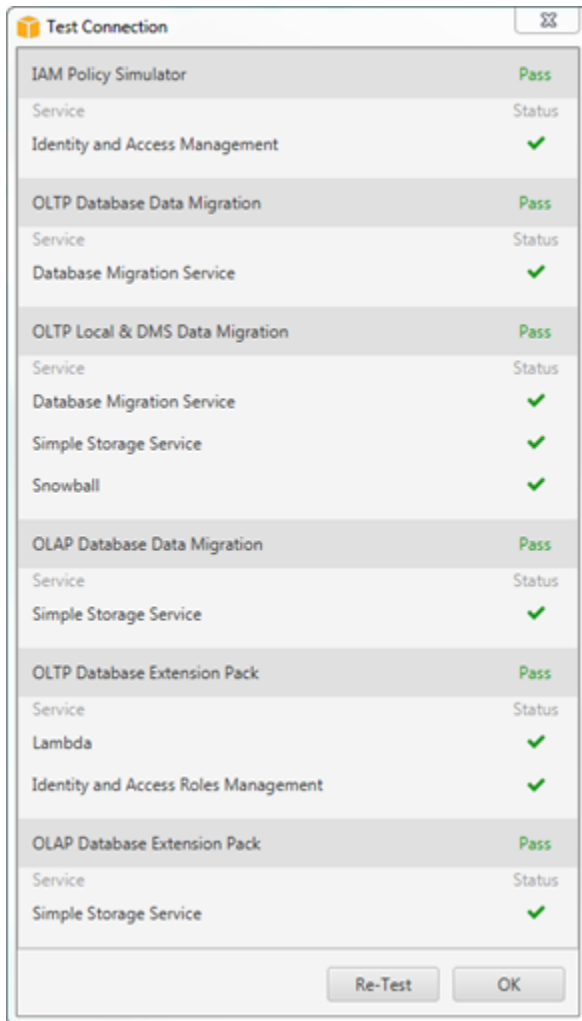
1. Mulai AWS Schema Conversion Tool.
2. Buka menu Pengaturan, lalu pilih Pengaturan global. Kotak dialog Pengaturan global muncul.
3. Pilih profil AWS layanan, lalu pilih Tambahkan profil AWS layanan baru.
4. Masukkan AWS informasi Anda sebagai berikut.

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

Pilih Gunakan FIPS titik akhir untuk S3 jika Anda perlu mematuhi persyaratan keamanan untuk Standar Pemrosesan Informasi Federal (FIPS). FIPSEndpoint tersedia di AWS Wilayah berikut:

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

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



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

Menyetel profil default untuk proyek

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

Untuk mengatur profil default untuk sebuah proyek

1. Mulai AWS Schema Conversion Tool dan buat proyek baru.
2. Pada menu Pengaturan, pilih Pengaturan proyek. Kotak dialog Pengaturan proyek muncul.

3. Pilih tab Lingkungan proyek.
4. Pilih Tambahkan profil AWS layanan baru untuk menambahkan profil baru. Kemudian untuk profil AWS layanan, pilih profil yang ingin Anda kaitkan dengan proyek.
5. Pilih OK untuk menutup kotak dialog Pengaturan proyek. Anda juga dapat memilih Batal untuk membatalkan perubahan Anda.

Izin untuk menggunakan profil AWS layanan

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

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

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

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

Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool

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

Untuk informasi selengkapnya tentang penggunaan AWS Secrets Manager, lihat [Apa itu AWS Secrets Manager?](#) dalam AWS Secrets Manager User Guide. Untuk informasi selengkapnya tentang menyimpan AWS profil, lihat [Mengelola Profil di AWS Schema Conversion Tool](#).

Untuk mengambil kredensial database dari Secrets Manager

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

AWS SCT mendukung rahasia yang memiliki struktur berikut.

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

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

Menyimpan kata sandi di AWS Schema Conversion Tool

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

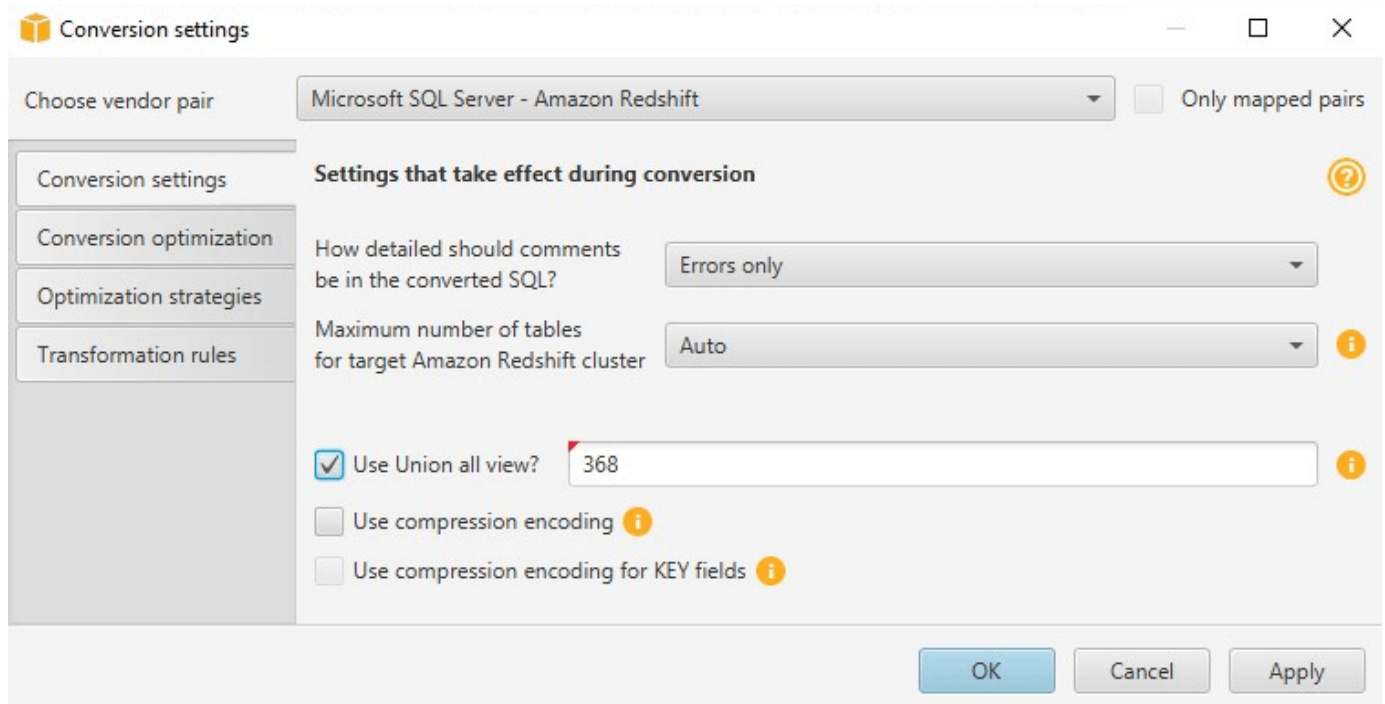
Kata sandi dienkripsi menggunakan token yang dihasilkan secara acak dalam file. `seed.dat`. Kata sandi kemudian disimpan dengan nama pengguna dalam file cache. Jika Anda kehilangan `seed.dat` file atau menjadi rusak, kata sandi database mungkin tidak dienkripsi secara tidak benar. Dalam hal ini, koneksi gagal.

Membuat UNION ALL tampilan di AWS Schema Conversion Tool

Jika tabel sumber dipartisi, AWS SCT membuat n tabel target, di mana n adalah jumlah partisi pada tabel sumber. AWS SCT membuat UNION ALL tampilan di atas tabel target untuk mewakili tabel sumber. Jika Anda menggunakan ekstraktor AWS SCT data untuk memigrasikan data Anda, partisi tabel sumber akan diekstraksi dan dimuat secara paralel oleh subtugas terpisah.

Untuk menggunakan tampilan Union All untuk sebuah proyek

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



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

Menggunakan Pintasan Keyboard di AWS Schema Conversion Tool

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

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

Pintasan keyboard	Deskripsi
Ctrl+F4	Tutup proyek yang terbuka.
F1	Buka Panduan AWS SCT Pengguna.

Memulai dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skema untuk database sumber. Basis data sumber dapat berupa mesin yang dikelola sendiri yang berjalan di lokasi atau di instans AmazonEC2. Anda dapat mengonversi skema sumber Anda ke skema untuk basis data apa pun yang didukung yang di-host oleh AWS. AWS SCT Aplikasi ini menyediakan antarmuka pengguna berbasis proyek.

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

1. Instal AWS SCT. Untuk informasi selengkapnya, lihat [Instalasi dan Konfigurasi AWS Schema Conversion Tool](#).
2. Instal AWS SCT agen, jika diperlukan. AWS SCT agen hanya diperlukan untuk skenario migrasi tertentu, seperti antara sumber dan target yang heterogen. Untuk informasi selengkapnya, lihat [Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool](#).
3. Biasakan diri Anda dengan antarmuka pengguna AWS SCT. Untuk informasi selengkapnya, lihat [Menavigasi antarmuka pengguna AWS SCT](#).
4. Buat AWS SCT proyek. Connect ke basis data sumber dan target. Untuk informasi selengkapnya tentang menghubungkan ke database sumber Anda, lihat [Menghubungkan ke database sumber dengan AWS Schema Conversion Tool](#).
5. Buat aturan pemetaan. Untuk informasi selengkapnya tentang aturan pemetaan, lihat [Memetakan tipe data di AWS Schema Conversion Tool](#).
6. Jalankan dan kemudian tinjau Laporan Penilaian Migrasi Database. Untuk informasi lebih lanjut tentang laporan penilaian, lihat [Melihat Laporan Penilaian di AWS Schema Conversion Tool](#).
7. Konversi skema basis data sumber. Ada beberapa aspek konversi yang perlu Anda ingat, seperti apa yang harus dilakukan dengan item yang tidak dikonversi, dan cara memetakan item yang harus dikonversi dengan cara tertentu. Untuk informasi selengkapnya tentang mengonversi skema sumber, lihat [Mengonversi skema database di AWS Schema Conversion Tool](#)

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

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

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

Anda juga dapat menggunakan AWS SCT untuk memigrasikan data Anda dari database sumber ke database yang dikelola Amazon. Sebagai contoh, lihat [Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool](#).

Menghubungkan ke database sumber dengan AWS Schema Conversion Tool

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

Informasi enkripsi

[Menghubungkan ke Amazon RDS dan dan Aurora yang dienkripsi](#)

Sumber database

- [Menghubungkan ke Apache Cassandra](#)
- [Menghubungkan ke Azure SQL](#)
- [Menghubungkan ke IBM DB2 untuk z/OS](#)
- [IBMDatabase Db2 LUW](#)
- [Mengggunakan My SQL sebagai sumber](#)
- [Database Oracle](#)
- [Database Postgre SQL](#)
- [SAPbasis data](#)
- [SQLDatabase server](#)

Sumber gudang data

- [Amazon Redshift](#)
- [Azure Synapse Analytics sebagai sumber](#)
- [BigQuery sebagai sumber](#)
- [Database Greenplum](#)
- [Database Netezza](#)
- [Gudang data Oracle](#)

- [Kepingan salju](#)
- [SQLGudang Data Server](#)
- [Database Teradata](#)
- [Database Vertica](#)

Sumber data besar

- [Menghubungkan ke Apache Hadoop](#)
- [Menghubungkan ke Apache Oozie](#)

Menghubungkan ke Amazon Relational Database Service terenkripsi dan database Amazon Aurora dengan AWS Schema Conversion Tool

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

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

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


Untuk mengimpor sertifikat atau sertifikat ke penyimpanan sistem Windows

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

Untuk informasi tentang mengunduh sertifikat, lihat [MenggunakanSSL/TLSuntuk mengenkripsi sambungan ke instans DB](#) di Panduan RDS Pengguna Amazon.

2. Di jendela pencarian Windows Anda, masukkan **Manage computer certificates**. Ketika diminta apakah akan membiarkan aplikasi membuat perubahan pada komputer Anda, pilih Ya.
3. Ketika jendela sertifikat terbuka, jika diperlukan perluas Sertifikat - Komputer Lokal sehingga Anda dapat melihat daftar sertifikat. Buka menu konteks (klik kanan) untuk Otoritas Sertifikasi Root Tepercaya, lalu pilih Semua Tugas, Impor.

- Pilih Berikutnya, lalu Jelajahi, dan temukan *.pem file yang Anda unduh di langkah 1. Pilih Buka untuk memilih file sertifikat, pilih Berikutnya, lalu pilih Selesai.

 Note

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

- Di Microsoft Management Console, perluas Sertifikat. Kemudian perluas Otoritas Sertifikasi Root Tepercaya, pilih Sertifikat, dan temukan sertifikat untuk mengonfirmasi bahwa sertifikat itu ada. Nama sertifikat dimulai dengan Amazon RDS.
- Mulai ulang komputer Anda.

Untuk mengimpor sertifikat atau sertifikat ke Java KeyStore

- Unduh sertifikat atau sertifikat dari salah satu sumber berikut:

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

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

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

Example

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

```
keytool -importcert -file eu-west-1-bundle.pem -alias eu-west-1-bundle.pem -
keystore trust-2019.ks
Picked up JAVA_TOOL_OPTIONS: -Dlog4j2.formatMsgNoLookups=true
Enter keystore password:
Re-enter new password:
```

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

Extensions:

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

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

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

#4: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 73 5F 60 D8 BC CB 03 98   F4 2B 17 34 2E 36 5A A6   s_`.....+.4.6Z.
0010: 60 FF BC 1F                               `...
]
]
```

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

4. Tambahkan keystore sebagai toko kepercayaan di AWS SCT. Untuk melakukannya, dari menu utama pilih Settings, Global settings, Security, Trust store, lalu pilih Select existing trust store.

Setelah menambahkan toko kepercayaan, Anda dapat menggunakannya untuk mengonfigurasi koneksi yang SSL diaktifkan saat Anda membuat AWS SCT koneksi ke database. Dalam dialog AWS SCT Connect to database, pilih Use SSL dan pilih trust store yang dimasukkan sebelumnya.

Menghubungkan ke database Apache Cassandra dengan AWS Schema Conversion Tool

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

Menghubungkan ke Apache Cassandra sebagai sumber

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

Untuk terhubung ke database sumber Apache Cassandra

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

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

Parameter	Tindakan
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.

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

Menghubungkan ke database Apache Hadoop dengan AWS Schema Conversion Tool

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

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

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

Topik

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

Prasyarat untuk menggunakan Apache Hadoop sebagai sumber

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

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

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

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

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

Izin untuk menggunakan Hive sebagai sumber

Izin yang diperlukan untuk pengguna sumber Hive adalah sebagai berikut:

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

Untuk meningkatkan kecepatan migrasi, sebaiknya Anda menjalankan pemadatan untuk tab sumber ACID -transaksional.

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

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

Izin untuk digunakan HDFS sebagai sumber

Izin yang diperlukan untuk HDFS sebagai sumber adalah sebagai berikut:

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

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

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

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

Izin untuk digunakan HDFS sebagai target

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

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

Menghubungkan ke Apache Hadoop sebagai sumber

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

Untuk terhubung ke Apache Hadoop di AWS SCT CLI

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

Unduh JDBC driver yang diperlukan dan tentukan lokasi tempat Anda menyimpan file. Untuk informasi selengkapnya, lihat [Menginstal JDBC driver untuk AWS Schema Conversion Tool](#).

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

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

Anda dapat menggunakan contoh ini dan contoh berikut di Windows.

3. Buat AWS SCT proyek baru.

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

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

4. Tambahkan cluster Hadoop sumber Anda ke proyek.

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

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

```
AddSourceCluster
  -name: 'HADOOP_SOURCE'
  -vendor: 'HADOOP'
  -host: 'hadoop_address'
  -port: '22'
  -user: 'hadoop_user'
  -password: 'hadoop_password'
  -useSSL: 'true'
  -privateKeyPath: 'c:\path\name.pem'
  -passPhrase: 'hadoop_passphrase'
/
```

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

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

Menghubungkan ke sumber Anda Hive dan layanan HDFS

Anda dapat terhubung ke sumber Anda Hive dan HDFS layanan dengan. AWS SCT CLI Untuk terhubung ke Apache Hive, gunakan JDBC driver Hive versi 2.3.4 atau lebih tinggi. Untuk informasi selengkapnya, lihat [Menginstal JDBC driver untuk AWS Schema Conversion Tool](#).

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

- Buat SSH terowongan baru.

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

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

- Gunakan SSH terowongan yang ada.

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

- Connect ke Hive dan HDFS layanan Anda secara langsung.

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

Untuk terhubung ke Hive dan HDFS di AWS SCT CLI

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

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

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

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

Contoh kode berikut terhubung ke layanan Hive Anda tanpa terowongan.

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

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

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

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

3. Tambahkan HDFS layanan sumber Anda ke proyek.

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

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

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

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

/

Kode berikut terhubung ke HDFS layanan Apache Anda tanpa terowongan.

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

/

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

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

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

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

Menghubungkan ke Amazon EMR sebagai target

Anda dapat terhubung ke EMR cluster Amazon target Anda dengan file AWS SCT CLI. Untuk melakukannya, Anda mengotorisasi lalu lintas masuk dan menggunakan SSH. Dalam hal ini, AWS SCT memiliki semua izin yang diperlukan untuk bekerja dengan EMR cluster Amazon Anda. Untuk informasi selengkapnya, lihat [Sebelum Anda menyambung dan Connect ke node utama menggunakan SSH](#) Panduan EMR Manajemen Amazon.

AWS SCT terhubung ke Amazon EMR Hive dengan pengguna hadoop cluster. Untuk terhubung ke Amazon EMR Hive, gunakan JDBC driver Hive versi 2.6.2.1002 atau lebih tinggi. Untuk informasi selengkapnya, lihat [Menginstal JDBC driver untuk AWS Schema Conversion Tool](#).

Untuk terhubung ke Amazon EMR di AWS SCT CLI

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

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

```
AddTargetCluster
  -name: 'HADOOP_TARGET'
  -vendor: 'AMAZON_EMR'
  -host: 'ec2-44-44-55-66.eu-west-1.EXAMPLE.amazonaws.com'
  -port: '22'
  -user: 'emr_user'
  -password: 'emr_password'
  -useSSL: 'true'
```

```
-privateKeyPath: 'c:\path\name.pem'  
-passPhrase: '1234567890abcdef0!'  
-s3Name: 'S3_TARGET'  
-accessKey: 'AKIAIOSFODNN7EXAMPLE'  
-secretKey: 'wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY'  
-region: 'eu-west-1'  
-s3Path: 'doc-example-bucket/example-folder'  
/  

```

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

3. Tambahkan bucket Amazon S3 target Anda ke proyek.

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

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

```

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

4. Tambahkan layanan Hive target Anda ke proyek.

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

```
AddTargetClusterHive  
-cluster: 'HADOOP_TARGET'  
-name: 'HIVE_TARGET'
```



```
-host: 'localhost'  
-port: '10006'  
-user: 'hive_user'  
-password: 'hive_password'  
-createTunnel: 'true'  
-localPort: '10006'  
-remoteHost: 'hive_address'  
-remotePort: 'hive_port'
```

```
/
```

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

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

5. Tambahkan HDFS layanan target Anda ke proyek.

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

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

```
/
```

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

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

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

Menghubungkan ke alur kerja Apache Oozie dengan AWS Schema Conversion Tool

Anda dapat menggunakan antarmuka baris AWS SCT perintah (CLI) untuk mengonversi alur kerja Apache Oozie menjadi AWS Step Functions. Setelah memigrasikan beban kerja Apache Hadoop Anda ke AmazonEMR, Anda dapat menggunakan layanan asli untuk mengatur pekerjaan Anda. AWS Cloud Untuk informasi selengkapnya, lihat [Menghubungkan ke Apache Hadoop](#).

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

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

Prasyarat untuk menggunakan Apache Oozie sebagai sumber

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

- Buat bucket Amazon S3 untuk menyimpan definisi mesin status. Anda dapat menggunakan definisi ini untuk mengonfigurasi mesin status Anda. Lihat informasi yang lebih lengkap di [Membuat bucket](#) dalam Panduan Pengguna Amazon S3.
- Buat peran AWS Identity and Access Management (IAM) dengan AmazonS3FullAccess kebijakan. AWS SCT menggunakan IAM peran ini untuk mengakses bucket Amazon S3 Anda.
- Catat kunci AWS rahasia dan kunci akses AWS rahasia Anda. Untuk informasi selengkapnya tentang kunci AWS akses, lihat [Mengelola kunci akses](#) di Panduan IAM Pengguna.
- Simpan AWS kredensial Anda dan informasi tentang bucket Amazon S3 Anda di profil layanan AWS di pengaturan aplikasi global. Kemudian, AWS SCT gunakan profil AWS layanan ini untuk bekerja dengan AWS sumber daya Anda. Untuk informasi selengkapnya, lihat [Mengelola Profil di AWS Schema Conversion Tool](#).

Untuk bekerja dengan alur kerja Apache Oozie sumber Anda, AWS SCT memerlukan struktur spesifik file sumber Anda. Setiap folder aplikasi Anda harus menyertakan `job.properties` file. File

ini mencakup pasangan nilai kunci dari properti pekerjaan Anda. Selain itu, setiap folder aplikasi Anda harus menyertakan `workflow.xml` file. File ini menjelaskan node tindakan dan node alur kontrol alur kerja Anda.

Menghubungkan ke Apache Oozie sebagai sumber

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

Untuk terhubung ke Apache Oozie di AWS SCT CLI

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

Contoh kode berikut menyimpan pengaturan aplikasi dan memungkinkan untuk menyimpan kata sandi dalam proyek Anda. Anda dapat menggunakan pengaturan yang disimpan ini di proyek lain.

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

3. Buat AWS SCT proyek baru.

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

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

4. Tambahkan folder dengan file Apache Oozie sumber Anda ke proyek menggunakan perintah. `AddSource` Pastikan Anda menggunakan `APACHE_OOZIE` nilai untuk `vendor` parameter. Juga, berikan nilai untuk parameter yang diperlukan berikut: `name` dan `mappingsFolder`.

Contoh kode berikut menambahkan Apache Oozie sebagai sumber dalam proyek Anda. AWS SCT Contoh ini membuat objek sumber dengan nama `OOZIE`. Gunakan nama objek ini untuk

menambahkan aturan pemetaan. Setelah Anda menjalankan contoh kode ini, AWS SCT gunakan `c:\oozie` folder untuk memuat file sumber Anda dalam proyek.

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

Anda dapat menggunakan contoh ini dan contoh berikut di Windows.

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

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

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

Izin untuk menggunakan AWS Lambda fungsi dalam paket ekstensi

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

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

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

```

```

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

```

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

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:ListRolePolicies",
                "iam:CreateRole",
                "iam:TagRole",
                "iam:PutRolePolicy",
                "iam>DeleteRolePolicy",
                "iam>DeleteRole",
                "iam:PassRole"
            ]
        }
    ]
}

```

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

Menghubungkan AWS Step Functions sebagai target

Gunakan prosedur berikut untuk terhubung AWS Step Functions sebagai target.

Untuk terhubung ke AWS Step Functions dalam AWS SCT CLI

1. Buka CLI skrip Anda yang mencakup informasi koneksi untuk file sumber Apache Oozie Anda.

2. Tambahkan informasi tentang target migrasi Anda dalam AWS SCT proyek menggunakan `AddTarget` perintah. Pastikan Anda menggunakan `STEP_FUNCTIONS` nilai untuk `vendor` parameter. Juga, berikan nilai untuk parameter yang diperlukan berikut: `name` dan `profile`.

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

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

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

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

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

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

4. Simpan CLI skrip Anda. Selanjutnya, tambahkan aturan pemetaan dan perintah migrasi. Untuk informasi selengkapnya, lihat [Mengonversi alur kerja Oozie](#);

Menghubungkan ke Microsoft Azure SQL Database dengan AWS SCT

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

- Amazon RDS untuk Saya SQL
- Amazon Aurora Edisi Kompatibel Saya SQL
- Amazon RDS para Postgre SQL
- Amazon Aurora SQL Postgre -Edisi Kompatibel

Topik

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

Hak istimewa untuk Azure SQL Database sebagai sumber

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

- VIEW DEFINITION
- VIEW DATABASE STATE

Ulangi hibah untuk setiap database yang skema Anda konversi.

Hak istimewa yang diperlukan untuk SQL database target My SQL dan Postgre dijelaskan di bagian berikut.

- [Hak istimewa untuk My SQL sebagai basis data target](#)
- [Hak istimewa untuk Postgre SQL sebagai basis data target](#)

Menghubungkan ke Azure SQL Database sebagai sumber

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

Untuk menyambung ke SQL database sumber Azure Database

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

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

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

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

Menghubungkan ke IBM DB2 untuk Database z/OS dengan AWS Schema Conversion Tool

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

- Amazon RDS untuk Saya SQL
- Amazon Aurora Edisi Kompatibel Saya SQL
- Amazon RDS para Postgre SQL
- Amazon Aurora SQL Postgre -Edisi Kompatibel

Prasyarat untuk Db2 untuk z/OS sebagai database sumber

IBMDB2 untuk z/OS versi 12 fungsi level 100 versi basis data tidak mendukung sebagian besar kemampuan baru IBM Db2 untuk z/OS versi 12. Versi database ini menyediakan dukungan untuk fallback ke Db2 versi 11 dan berbagi data dengan Db2 versi 11. Untuk menghindari konversi fitur Db2 versi 11 yang tidak didukung, kami sarankan Anda menggunakan IBM Db2 untuk fungsi database z/OS tingkat 500 atau lebih tinggi sebagai sumber untuk AWS SCT

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

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

Pastikan kode ini mengembalikan versi DSN12015 atau lebih tinggi.

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

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

Pastikan kode ini mengembalikan versi V12R1M500 atau lebih tinggi.

Hak istimewa untuk Db2 untuk z/OS sebagai database sumber

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

- SELECTPADASYSIBM. LOCATIONS
- SELECTPADASYSIBM. SYSCHECKS
- SELECTPADASYSIBM. SYSCOLUMNS
- SELECTPADASYSIBM. SYSDATABASE
- SELECTPADASYSIBM. SYSDATATYPES
- SELECTPADASYSIBM. SYSDUMMY1
- SELECTPADASYSIBM. SYSFORIGNKEYS
- SELECTPADASYSIBM. SYSINDEXES
- SELECTPADASYSIBM. SYSKEYCOLUSE
- SELECTPADASYSIBM. SYSKEYS
- SELECTPADASYSIBM. SYSKEYTARGETS
- SELECTPADASYSIBM. SYSJAROBJECTS
- SELECTPADASYSIBM. SYSPACKAGE
- SELECTPADASYSIBM. SYSPARMS
- SELECTPADASYSIBM. SYSRELS
- SELECTPADASYSIBM. SYSROUTINES
- SELECTPADASYSIBM. SYSSEQUENCES
- SELECTPADASYSIBM. SYSSEQUENCESDEP

- SELECTPADASYSIBM. SYSSYNONYMS
- SELECTPADASYSIBM. SYSTABCONST
- SELECTPADASYSIBM. SYSTABLES
- SELECTPADASYSIBM. SYSTABLESPACE
- SELECTPADASYSIBM. SYSTRIGGERS
- SELECTPADASYSIBM. SYSVARIABLES
- SELECTPADASYSIBM. SYSVIEWS

Untuk mengonversi Db2 untuk tabel z/OS ke tabel SQL partisi Postgre, kumpulkan statistik pada ruang tabel dan tabel di database Anda menggunakan utilitas seperti yang ditunjukkan berikut.

RUNSTATS

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

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

Menghubungkan ke Db2 untuk z/OS sebagai sumber

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

Untuk terhubung ke IBM Db2 untuk database sumber z/OS

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

Kotak dialog Tambah sumber muncul.

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

1. Untuk AWS Rahasia, pilih nama rahasianya.
2. Pilih Isi untuk secara otomatis mengisi semua nilai di kotak dialog koneksi database dari Secrets Manager.

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

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

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

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

Hak istimewa untuk My SQL sebagai basis data target

Keistimewaan yang diperlukan untuk My SQL sebagai target adalah sebagai berikut:

- CREATEPADA * . *
- ALTERPADA * . *
- DROPPADA * . *

- INDEXPADA * . *
- REFERENCESPADA * . *
- SELECTPADA * . *
- CREATEVIEWPADA * . *
- SHOWVIEWPADA * . *
- TRIGGERPADA * . *
- CREATEROUTINEPADA * . *
- ALTERROUTINEPADA * . *
- EXECUTEPADA * . *
- SELECTPADA mysql.proc
- INSERT, UPDATE PADA AWS _ DB2ZOS _ EXT. *
- INSERT, UPDATE, DELETE PADA AWS _ DB2ZOS _ EXT _ DATA. *
- CREATETEMPORARYTABLESPADA AWS _ DB2ZOS _ EXT _ DATA. *

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

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

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

Untuk menggunakan Amazon RDS for My SQL sebagai target, atur `log_bin_trust_function_creators` parameter ke `true`, dan `character_set_server` ke `latin1`. Untuk mengkonfigurasi parameter ini, buat grup parameter DB baru atau modifikasi grup parameter DB yang ada.

Untuk menggunakan Aurora My SQL sebagai target, atur `log_bin_trust_function_creators` parameter ke `true`, dan `to. character_set_server latin1` Juga, atur `lower_case_table_names` parameter ke `true`. Untuk mengkonfigurasi parameter ini, buat grup parameter DB baru atau modifikasi grup parameter DB yang ada.

Hak istimewa untuk Postgre SQL sebagai basis data target

Untuk menggunakan Postgre SQL sebagai target, AWS SCT membutuhkan hak istimewa. `CREATE ON DATABASE` Pastikan Anda memberikan hak istimewa ini untuk setiap basis data SQL Postgre target.

Untuk menggunakan Amazon RDS untuk Postgre SQL sebagai target, AWS SCT membutuhkan hak istimewa. `rds_superuser`

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

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

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

Pada contoh sebelumnya, ganti `user_name` dengan nama pengguna Anda. Kemudian, ganti `db_name` dengan nama database target Anda. Akhirnya, ganti `your_password` dengan kata sandi yang aman.

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

Saat Anda menggunakan pengguna yang berbeda untuk mengonversi dan menerapkan skema yang berbeda ke basis data target Anda, Anda bisa mendapatkan pesan kesalahan saat tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan kesalahan ini, gunakan `superuser` peran.

Db2 untuk z/OS ke pengaturan konversi Postgre SQL

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

Db2 untuk pengaturan SQL konversi z/OS ke Postgre di AWS SCT menyertakan opsi untuk yang berikut:

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

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

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

- Untuk menghasilkan nama unik untuk kendala dalam database target.

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

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

AWS SCT dapat menyimpan tata letak nama kolom, ekspresi, dan klausa dalam DML pernyataan dalam posisi dan urutan yang sama seperti pada kode sumber. Untuk melakukannya, pilih Ya untuk Pertahankan pemformatan nama kolom, ekspresi, dan klausa dalam pernyataan. DML

- Untuk mengecualikan partisi tabel dari lingkup konversi.

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

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

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

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

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

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

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

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

- Organisasi Standar Internasional (ISO)
- IBMStandar Eropa (EUR)
- IBMUSASandar (USA)
- Era Kristen Standar Industri Jepang (JIS)

ConnConnecting to IBM DB2 untuk Linux,UNIX, dan Windows Database dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode dalam SQL bahasa, dan kode aplikasi dari IBM Db2 untuk Linux, Unix, dan Windows (Db2LUW) ke target berikut.

- Amazon RDS untuk Saya SQL
- Amazon Aurora Edisi Kompatibel Saya SQL
- Amazon RDS para Postgre SQL

- Amazon Aurora SQL Postgre -Edisi Kompatibel
- Amazon RDS untuk MariaDB

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

Hak istimewa untuk Db2 LUW sebagai sumber

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

- Hak istimewa yang diperlukan untuk membuat koneksi:
 - CONNECTPADA DATABASE
- Hak istimewa diperlukan untuk menjalankan SQL pernyataan:
 - EXECUTE PADA PACKAGENULLID. SYSSH200
- Hak istimewa yang diperlukan untuk mendapatkan informasi tingkat instans:
 - EXECUTE PADA FUNCTIONSYSPROC. ENV_GET _ INST _ INFO
 - SELECT PADA SYSIBMADM. ENV_ INST _ INFO
 - SELECT PADA SYSIBMADM. ENV_ SYS _ INFO
- Hak istimewa yang diperlukan untuk memeriksa hak istimewa yang diberikan melalui peran, kelompok, dan otoritas:
 - EXECUTE PADA FUNCTIONSYSPROC. AUTH_ LIST _ AUTHORITIES _ FOR _ AUTHID
 - EXECUTE PADA FUNCTIONSYSPROC. AUTH_ LIST _ GROUPS _ FOR _ AUTHID
 - EXECUTE PADA FUNCTIONSYSPROC. AUTH_ LIST _ ROLES _ FOR _ AUTHID
 - SELECT PADA SYSIBMADM. PRIVILEGES
- Hak istimewa yang diperlukan pada katalog dan tabel sistem:
 - SELECT PADA SYSCAT. ATTRIBUTES
 - SELECT PADA SYSCAT. CHECKS
 - SELECT PADA SYSCAT. COLIDENTATTRIBUTES
 - SELECT PADA SYSCAT. COLUMNS
 - SELECT PADA SYSCAT. DATAPARTITIONEXPRESSION
 - SELECT PADA SYSCAT. DATAPARTITIONS
 - SELECT PADA SYSCAT. DATATYPEDEP
 - SELECT PADA SYSCAT. DATATYPES

- SELECTPADASYSCAT. HIERARCHIES
 - SELECTPADASYSCAT. INDEXCOLUSE
 - SELECTPADASYSCAT. INDEXES
 - SELECTPADASYSCAT. INDEXPARTITIONS
 - SELECTPADASYSCAT. KEYCOLUSE
 - SELECTPADASYSCAT. MODULEOBJECTS
 - SELECTPADASYSCAT. MODULES
 - SELECTPADASYSCAT. NICKNAMES
 - SELECTPADASYSCAT. PERIODS
 - SELECTPADASYSCAT. REFERENCES
 - SELECTPADASYSCAT. ROUTINEPARMS
 - SELECTPADASYSCAT. ROUTINES
 - SELECTPADASYSCAT. ROWFIELDS
 - SELECTPADASYSCAT. SCHEMATA
 - SELECTPADASYSCAT. SEQUENCES
 - SELECTPADASYSCAT. TABCONST
 - SELECTPADASYSCAT. TABLES
 - SELECTPADASYSCAT. TRIGGERS
 - SELECTPADASYSCAT. VARIABLEDEP
 - SELECTPADASYSCAT. VARIABLES
 - SELECTPADASYSCAT. VIEWS
 - SELECTPADASYSCAT. SYSDUMMY1
- Untuk menjalankan SQL pernyataan, akun pengguna memerlukan hak istimewa untuk menggunakan setidaknya salah satu beban kerja yang diaktifkan dalam database. Jika tidak ada beban kerja yang ditetapkan ke pengguna, pastikan bahwa beban kerja pengguna default dapat diakses oleh pengguna:
 - USAGEPADA WORKLOAD SYSDEFAULTUSERWORKLOAD

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

```
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;
```

Menghubungkan ke Db2 LUW sebagai sumber

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

Untuk terhubung ke database sumber Db2 LUW

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

Parameter	Tindakan
Nama server	Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis Data	Masukkan nama database Db2LUW.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>

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

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

Migrasi dari IBM DB2 LinuxUNIX, dan Windows ke Amazon Relational Database Service untuk SQL Postgre atau Amazon Aurora Postgre - Compatible Edition SQL

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

- Peristiwa pemicu — INSERTDELETE, dan peristiwa UPDATE pemicu menentukan bahwa tindakan yang dipicu berjalan setiap kali acara diterapkan ke tabel subjek atau tampilan subjek. Anda dapat menentukan kombinasi dari INSERT, DELETE, dan UPDATE peristiwa, tetapi Anda dapat menentukan setiap acara hanya sekali. AWS SCT mendukung peristiwa pemicu tunggal dan ganda. Untuk acara, Postgre praktis SQL memiliki fungsi yang sama.
- Event OF COLUMN - Anda dapat menentukan nama kolom dari tabel dasar. Pemicu diaktifkan hanya dengan pembaruan kolom yang diidentifikasi dalam daftar nama kolom. Postgre SQL memiliki fungsi yang sama.
- Pemicu pernyataan — Ini menentukan bahwa tindakan yang dipicu diterapkan hanya sekali untuk seluruh pernyataan. Anda tidak dapat menentukan jenis granularitas pemicu ini untuk BEFORE pemicu atau pemicu INSTEAD OF. Jika ditentukan, DELETE pemicu UPDATE atau diaktifkan, bahkan jika tidak ada baris yang terpengaruh. Postgre SQL juga memiliki fungsi ini dan memicu deklarasi untuk pemicu pernyataan identik untuk SQL Postgre dan Db2. LUW
- Klausula referensi - Ini menentukan nama korelasi untuk variabel transisi dan nama tabel untuk tabel transisi. Nama korelasi mengidentifikasi baris tertentu dalam kumpulan baris yang dipengaruhi oleh operasi pemicu. SQL Nama tabel mengidentifikasi set lengkap baris yang terpengaruh. Setiap baris yang terpengaruh oleh SQL operasi pemicu tersedia untuk tindakan yang dipicu dengan mengkualifikasi kolom dengan nama korelasi tertentu. Postgre SQL tidak mendukung fungsi ini, dan hanya menggunakan nama NEW atau OLD korelasi.
- INSTEADOF pemicu - AWS SCT mendukung ini.

Mengonversi tabel LUW dipartisi Db2 ke Postgre versi 10 tabel dipartisi SQL

AWS SCT dapat mengonversi tabel Db2 ke LUW tabel yang dipartisi di Postgre 10. SQL Ada beberapa batasan saat mengonversi tabel LUW partisi Db2 ke Postgre: SQL

- Anda dapat membuat tabel yang dipartisi dengan kolom nullable di Db2LUW, dan Anda dapat menentukan partisi untuk menyimpan nilai. NULL Namun, Postgre SQL tidak mendukung NULL nilai untuk RANGE partisi.
- Db2 LUW dapat menggunakan INCLUSIVE atau EXCLUSIVE klausa untuk mengatur nilai batas rentang. Postgre SQL hanya mendukung INCLUSIVE batas awal dan batas akhirEXCLUSIVE. Nama partisi yang dikonversi dalam format <original_table_name>_<original_partition_name>.
- Anda dapat membuat kunci primer atau unik untuk tabel yang dipartisi di Db2. LUW Postgre SQL mengharuskan Anda untuk membuat kunci primer atau unik untuk setiap partisi secara langsung. Batasan kunci primer atau unik harus dihapus dari tabel induk. Nama kunci yang dikonversi ada dalam format <original_key_name>_<original_partition_name>.

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

Hak istimewa untuk Postgre SQL sebagai target

Untuk menggunakan Postgre SQL sebagai target, AWS SCT membutuhkan hak istimewa. `CREATE ON DATABASE` Pastikan Anda memberikan hak istimewa ini untuk setiap basis data SQL Postgre target.

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

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

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

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

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

Saat Anda menggunakan pengguna yang berbeda untuk mengonversi dan menerapkan skema yang berbeda ke basis data target Anda, Anda bisa mendapatkan pesan kesalahan saat tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan kesalahan ini, gunakan `superuser` peran.

Bermigrasi dari IBM DB2 untuk Linux,UNIX, dan Windows ke Amazon RDS untuk My SQL atau Amazon Aurora My SQL

Saat Anda mengonversi LUW database IBM Db2 ke RDS for My SQL atau Amazon Aurora SQL My, perhatikan hal-hal berikut.

Hak istimewa untuk saya SQL sebagai target

Keistimewaan yang diperlukan untuk My SQL sebagai target adalah sebagai berikut:

- CREATEPADA * . *
- ALTERPADA * . *
- DROPPADA * . *
- INDEXPADA * . *
- REFERENCESPADA * . *
- SELECTPADA * . *
- CREATEVIEWPADA * . *
- SHOWVIEWPADA * . *
- TRIGGERPADA * . *
- CREATEROUTINEPADA * . *
- ALTERROUTINEPADA * . *
- EXECUTEPADA * . *
- SELECTPADA mysql.proc
- INSERT, UPDATE PADA AWS _ DB2 _ EXT. *
- INSERT,UPDATE, DELETE PADA AWS _ DB2 _ EXT _ DATA. *
- CREATETEMPORARYTABLESPADA AWS _ DB2 _ EXT _ DATA. *

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

```
CREATE USER 'user_name' IDENTIFIED BY 'your_password';  
GRANT CREATE ON *.* TO 'user_name';  
GRANT ALTER ON *.* TO 'user_name';
```

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

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

Untuk menggunakan Amazon RDS untuk My SQL atau Aurora My SQL sebagai target, tetapkan `lower_case_table_names` parameternya ke 1. Nilai ini berarti bahwa SQL server Saya menangani pengidentifikasi nama objek seperti tabel, indeks, pemacu, dan database sebagai tidak peka huruf besar/kecil. Jika Anda telah mengaktifkan logging biner dalam instance target Anda, maka atur `log_bin_trust_function_creators` parameternya ke 1. Dalam hal ini, Anda tidak perlu menggunakan `DETERMINISTIC`, `READS SQL DATA` atau `NO SQL` karakteristik untuk membuat fungsi yang disimpan. Untuk mengkonfigurasi parameter ini, buat grup parameter DB baru atau modifikasi grup parameter DB yang ada.

Menggunakan My SQL sebagai sumber untuk AWS SCT

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode database, dan kode aplikasi dari My SQL ke target berikut:

- Amazon RDS para Postgre SQL
- Amazon Aurora SQL Postgre -Edisi Kompatibel
- Amazon RDS untuk Saya SQL

Untuk informasi selengkapnya, lihat bagian berikut:

Topik

- [Hak istimewa untuk My SQL sebagai database sumber](#)
- [Menghubungkan ke My SQL sebagai sumber](#)
- [Hak istimewa untuk Postgre SQL sebagai basis data target](#)

Hak istimewa untuk My SQL sebagai database sumber

Keistimewaan yang diperlukan untuk My SQL sebagai sumber adalah sebagai berikut:

- SELECTPADA * . *
- SHOWVIEWPADA * . *

Menghubungkan ke My SQL sebagai sumber

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

Untuk terhubung ke database SQL sumber saya

1. Di dalam AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Milik Saya SQL, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

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

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

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

Parameter	Tindakan
Nama server	<p>Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.</p> <p>Anda dapat terhubung ke sumber Anda SQL Database saya menggunakan protokol IPv6 alamat. Untuk melakukannya, pastikan Anda menggunakan tanda kurung siku untuk memasukkan alamat IP, seperti yang ditunjukkan pada contoh berikut.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"><code>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</code></div>
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>

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

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

Hak istimewa untuk Postgre SQL sebagai basis data target

Untuk menggunakan Postgre SQL sebagai target, AWS SCT membutuhkan hak istimewa. CREATE ON DATABASE Pastikan Anda memberikan hak istimewa ini untuk setiap basis data SQL Postgre target.

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

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

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

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

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

Saat Anda menggunakan pengguna yang berbeda untuk mengonversi dan menerapkan skema yang berbeda ke basis data target Anda, Anda bisa mendapatkan pesan kesalahan saat tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan kesalahan ini, gunakan superuser peran.

Menghubungkan ke Database Oracle dengan AWS Schema Conversion Tool

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

- Amazon RDS untuk Saya SQL
- Amazon Aurora Edisi Kompatibel Saya SQL
- Amazon RDS para Postgre SQL
- Amazon Aurora SQL Postgre -Edisi Kompatibel
- Amazon RDS untuk Oracle
- Amazon RDS untuk MariaDB

Ketika sumbernya adalah database Oracle, komentar dapat dikonversi ke format yang sesuai, misalnya, database PostgreSQL. AWS SCT dapat mengkonversi komentar pada tabel, tampilan, dan kolom. Komentar dapat menyertakan apostrof; AWS SCT menggandakan apostrof saat mengonversi SQL pernyataan, seperti halnya untuk literal string.

Untuk informasi selengkapnya, lihat hal berikut.

Topik

- [Hak istimewa untuk Oracle sebagai sumber](#)
- [Menghubungkan ke Oracle sebagai sumber](#)
- [Bermigrasi dari Oracle ke Amazon RDS untuk Postgre atau Amazon SQL Aurora Postgre dengan SQL AWS Schema Conversion Tool](#)
- [Bermigrasi dari Oracle ke Amazon RDS untuk My SQL atau Amazon SQL Aurora My dengan AWS Schema Conversion Tool](#)
- [Bermigrasi dari Oracle Database ke Amazon RDS untuk Oracle dengan AWS Schema Conversion Tool](#)

Hak istimewa untuk Oracle sebagai sumber

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

- CONNECT
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY
- SELECTPADASYS. ARGUMENT\$

Menghubungkan ke Oracle sebagai sumber

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

Untuk terhubung ke database sumber Oracle

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

Parameter	Tindakan
Jenis	<p>Pilih jenis koneksi ke database Anda. Tergantung pada jenis Anda, berikan informasi tambahan berikut:</p> <ul style="list-style-type: none">• SID<ul style="list-style-type: none">• Nama server: Nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.• Port server: Port yang digunakan untuk terhubung ke server database sumber Anda.

Parameter	Tindakan
	<ul style="list-style-type: none"> • Oracle SID: ID Sistem Oracle (SID). Untuk menemukan OracleSID, kirimkan kueri berikut ke database Oracle Anda: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre> • Nama layanan • Nama server: DNS Nama atau alamat IP server basis data sumber Anda. <p>Anda dapat terhubung ke database Oracle sumber Anda menggunakan protokol IPv6 alamat. Untuk melakukannya, pastikan Anda menggunakan tanda kurung siku untuk memasukkan alamat IP, seperti yang ditunjukkan pada contoh berikut.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"> <p>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</p> </div> • Port server: Port yang digunakan untuk terhubung ke server database sumber Anda. • Nama layanan: Nama layanan Oracle untuk terhubung. • TNSalias • TNSpath file: Path ke file yang berisi informasi koneksi nama Transparent Network Substrate (TNS). <p>Setelah Anda memilih TNS file, AWS SCT tambahkan semua koneksi database Oracle dari file ke daftar TNSalias.</p> <p>Pilih opsi ini untuk terhubung ke Oracle Real Application Clusters (RAC).</p> • TNSalias: TNS Alias dari file ini untuk digunakan untuk terhubung ke database sumber. • TNSsambungkan pengenal

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

Parameter	Tindakan
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Pilih opsi ini untuk menyimpan kata sandi database dan untuk terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.
Jalur pengemudi Oracle	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

Bermigrasi dari Oracle ke Amazon RDS untuk Postgre atau Amazon SQL Aurora Postgre dengan SQL AWS Schema Conversion Tool

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

Topik

- [Hak istimewa untuk Postgre SQL sebagai basis data target](#)
- [Pengaturan konversi Oracle ke Postgre SQL](#)
- [Mengonversi urutan Oracle](#)
- [Mengonversi Oracle ROWID](#)
- [Mengonversi Oracle Dynamic SQL](#)
- [Mengonversi partisi Oracle](#)

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

Objek sistem Oracle	Deskripsi	Objek Postgre SQL yang dikonversi
V \$ VERSION	Menampilkan nomor versi komponen pustaka inti di Oracle Database	aws_oracle_ext.v \$versi
V \$ INSTANCE	Tampilan yang menunjukkan keadaan instance saat ini.	aws_oracle_ext.v \$contoh

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

Hak istimewa untuk Postgre SQL sebagai basis data target

Untuk menggunakan Postgre SQL sebagai target, AWS SCT membutuhkan hak istimewa. CREATE ON DATABASE Pastikan Anda memberikan hak istimewa ini untuk setiap basis data SQL Postgre target.

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

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

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

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

Untuk menggunakan Amazon RDS untuk Postgre SQL sebagai target, AWS SCT membutuhkan hak istimewa. `rds_superuser`

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

Saat Anda menggunakan pengguna yang berbeda untuk mengonversi dan menerapkan skema yang berbeda ke basis data target Anda, Anda bisa mendapatkan pesan kesalahan saat tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan kesalahan ini, gunakan `superuser` peran.

Pengaturan konversi Oracle ke Postgre SQL

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

Pengaturan SQL konversi Oracle ke Postgre AWS SCT termasuk opsi untuk yang berikut:

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

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

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

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

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

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

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

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

Mengonversi urutan Oracle

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

Untuk mengisi urutan yang dikonversi dengan nilai terakhir dari database sumber

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

Mengonversi Oracle ROWID

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

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

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

Untuk membuat kolom data untuk Oracle ROWID untuk sebuah proyek

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

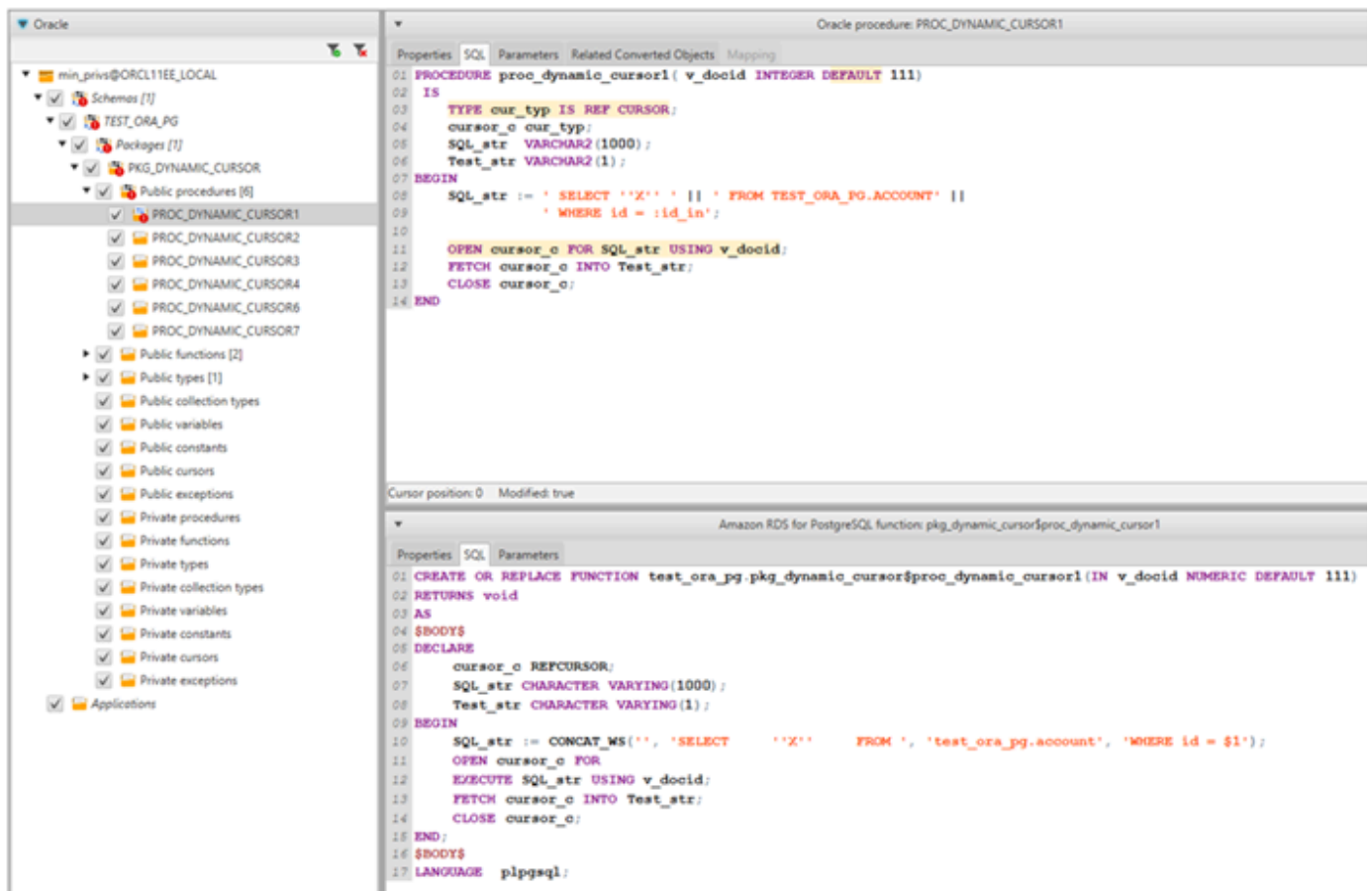
3. Dari daftar atas, pilih Oracle, dan kemudian pilih Oracle - Postgre. SQL AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Oracle ke PostgreSQL.
4. Untuk Menghasilkan ID baris, lakukan salah satu hal berikut:
 - Pilih Hasilkan sebagai identitas untuk membuat kolom data numerik.
 -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

Mengonversi Oracle Dynamic SQL

Oracle menyediakan dua cara untuk menerapkan dinamisSQL: menggunakan EXECUTE IMMEDIATE pernyataan atau prosedur panggilan dalam SQL paket DBMS _. Jika database Oracle sumber Anda menyertakan objek dengan dinamisSQL, gunakan AWS SCT untuk mengonversi SQL pernyataan dinamis Oracle ke Postgre. SQL

Untuk mengonversi Oracle dynamic SQL ke Postgre SQL

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



Mengonversi partisi Oracle

AWS SCT saat ini mendukung metode partisi berikut:

- Kisaran
- Daftar
- Rentang multicolumn
- Hash
- Komposit (daftar-daftar, daftar rentang, daftar-rentang, daftar-hash, rentang-hash, hash hash)

Bermigrasi dari Oracle ke Amazon RDS untuk My SQL atau Amazon SQL Aurora My dengan AWS Schema Conversion Tool

Untuk meniru fungsi database Oracle dalam SQL kode Saya yang dikonversi, gunakan paket SQL ekstensi Oracle to My di. AWS SCT Untuk informasi selengkapnya tentang paket ekstensi, lihat [Menggunakan paket ekstensi dengan AWS Schema Conversion Tool](#).

Topik

- [Hak istimewa untuk My SQL sebagai database target](#)
- [Oracle ke pengaturan SQL konversi Saya](#)
- [Pertimbangan migrasi](#)
- [Mengonversi WITH pernyataan di Oracle menjadi RDS untuk My SQL atau Amazon Aurora My SQL](#)

Hak istimewa untuk My SQL sebagai database target

Hak istimewa yang diperlukan untuk My SQL sebagai target adalah sebagai berikut:

- CREATEPADA * . *
- ALTERPADA * . *
- DROPPADA * . *
- INDEXPADA * . *
- REFERENCESPADA * . *
- SELECTPADA * . *
- CREATEVIEWPADA * . *
- SHOWVIEWPADA * . *
- TRIGGERPADA * . *
- CREATEROUTINEPADA * . *
- ALTERROUTINEPADA * . *
- EXECUTEPADA * . *
- CREATETEMPORARYTABLESPADA * . *
- AWS_LAMBDA_ACCESS
- INSERT, UPDATE PADA AWS _ ORACLE _EXT. *

- INSERT,UPDATE, DELETE PADA AWS _ ORACLE _ EXT _ DATA. *

Jika Anda menggunakan SQL database Saya versi 5.7 atau lebih rendah sebagai target, berikan izin INVOKE LAMBDA *.* alih-alih __. AWS LAMBDA ACCESS Untuk SQL database Saya versi 8.0 dan yang lebih tinggi, berikan izin AWS _ LAMBDA _ ACCESS.

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

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

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

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

Untuk menggunakan Amazon RDS untuk My SQL atau Aurora My SQL sebagai target, tetapkan `lower_case_table_names` parameter ke 1 Nilai ini berarti bahwa SQL server Saya menangani pengidentifikasi nama objek seperti tabel, indeks, pemicu, dan database sebagai tidak peka huruf besar/kecil. Jika Anda telah mengaktifkan logging biner dalam instance target Anda, maka atur `log_bin_trust_function_creators` parameter ke 1. Dalam hal ini, Anda tidak perlu menggunakan DETERMINISTIC, READS SQL DATA atau NO SQL karakteristik untuk membuat fungsi

yang disimpan. Untuk mengonfigurasi parameter ini, buat grup parameter DB baru atau modifikasi grup parameter DB yang ada.

Oracle ke pengaturan SQL konversi Saya

Untuk mengedit Oracle ke pengaturan SQL konversi saya, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Oracle, lalu pilih Oracle — My. SQL AWS SCT menampilkan semua pengaturan yang tersedia untuk Oracle ke SQL konversi Saya.

Pengaturan SQL konversi Oracle ke My AWS SCT termasuk opsi untuk yang berikut:

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

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

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

- Untuk mengatasi bahwa database Oracle sumber Anda dapat menggunakan ROWID pseudocolumn tetapi My SQL tidak mendukung fungsionalitas serupa. AWS SCT dapat meniru ROWID pseudocolumn dalam kode yang dikonversi. Untuk melakukannya, pilih Hasilkan sebagai identitas untuk Hasilkan ID baris? .

Jika kode Oracle sumber Anda tidak menggunakan ROWID pseudocolumn, pilih Don't generate for Generate row ID? Dalam hal ini, kode yang dikonversi bekerja lebih cepat.

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

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

- Fungsi TO_CHAR () tidak menggunakan string pemformatan khusus Oracle
- Fungsi TO_DATE () tidak menggunakan string pemformatan khusus Oracle

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

Pertimbangan migrasi

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

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

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

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

Pernyataan Oracle	SQLPernyataan saya
<pre> BEGIN statement1; label1: statement2; GOTO label1; statement3; statement4; END </pre>	<pre> BEGIN statement1; label1: LOOP statement2; ITERATE label1; LEAVE label1; END LOOP; statement3; statement4; END </pre>
<pre> BEGIN statement1; label1: statement2; statement3; statement4; END </pre>	<pre> BEGIN statement1; label1: BEGIN statement2; statement3; statement4; END; END </pre>

Mengonversi WITH pernyataan di Oracle menjadi RDS untuk My SQL atau Amazon Aurora My SQL

Anda menggunakan WITH klausa (`subquery_factoring`) di Oracle untuk menetapkan nama (`query_name`) ke blok subquery. Anda kemudian dapat mereferensikan blok subquery beberapa

tempat dalam kueri dengan menentukan `query_name`. Jika blok subquery tidak berisi link atau parameter (lokal, prosedur, fungsi, paket), kemudian AWS SCT mengkonversi klausa ke tampilan atau tabel sementara.

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

Anda dapat menemukan contoh dalam tabel berikut.

Pernyataan Oracle	SQLPernyataan saya
<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;</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</pre>

Pernyataan Oracle

```
END LOOP;
```

SQLPernyataan saya

```
        FETCH cur INTO var$id, var
        $state;

        IF done THEN
            LEAVE read_label;
        END IF;

        BEGIN
            END;
        END LOOP;
        CLOSE cur;
    END;
```

Pernyataan Oracle

```

CREATE PROCEDURE
  TEST_ORA_PG.P_WITH_SELECT_R
  EGULAR_MULT_01
AS
BEGIN

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

```

SQLPernyataan saya

```

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

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

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

```

Pernyataan Oracle	SQLPernyataan saya
	<pre>WHERE dept_emp1.DEPT_ID = dept.DEPTID; DECLARE CONTINUE HANDLER FOR NOT FOUND SET done := TRUE; OPEN cur; read_label: LOOP FETCH cur INTO var\$ID, var\$NAME, var\$SURNAME, var\$LASTNAME, var\$STATE, var \$DEPT_ID, var\$deptid, var\$PARENT_ID, var\$deptname; IF done THEN LEAVE read_label; END IF; BEGIN END; END LOOP; CLOSE cur; END; call test_ora_pg.P_WITH_SELECT_R EGULAR_MULT_01()</pre>

Pernyataan Oracle

```

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

```

SQLPernyataan saya

```

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

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

  read_label:

```

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

Bermigrasi dari Oracle Database ke Amazon RDS untuk Oracle dengan AWS Schema Conversion Tool

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

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

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

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

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

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

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

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

Ada beberapa prasyarat untuk menggunakan Oracle Spatial dan Oracle Locator di Amazon untuk instance Oracle DB: RDS

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

- Instance harus kelas instans DB yang dapat mendukung fitur Oracle. Misalnya, Oracle Spatial tidak didukung untuk kelas instans db.m1.small, db.t1.micro, db.t2.micro, atau db.t2.small DB. Untuk informasi selengkapnya, lihat [dukungan kelas instans DB untuk Oracle](#).
- Instans harus mengaktifkan opsi Auto Minor Version Upgrade. Amazon RDS memperbarui instans DB Anda ke Oracle terbaru PSU jika ada kerentanan keamanan dengan CVSS skor 9+ atau kerentanan keamanan lainnya yang diumumkan. Untuk informasi selengkapnya, silakan lihat

[Pengaturan untuk instans Oracle DB.](#)

- Jika instans DB Anda adalah versi 11.2.0.4.v10 atau lebih tinggi, Anda harus menginstal opsi XMLDB Untuk informasi selengkapnya, silakan lihat

[Oracle XML DB.](#)

- Anda harus memiliki lisensi Oracle Spatial dari Oracle. Untuk informasi selengkapnya, lihat [Oracle Spatial and Graph](#) dalam dokumentasi Oracle.
- Data Guard disertakan dengan Oracle Database Enterprise Edition. Untuk ketersediaan tinggi, gunakan fitur Amazon RDS Multi-AZ.

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

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

Setelah Anda membuat tautan database, Anda dapat menggunakan tautan dalam SQL pernyataan untuk merujuk ke tabel, tampilan, dan SQL objek PL/di database lainnya. Untuk menggunakan link database, tambahkan @dblink ke tabel, tampilan, atau nama SQL objek PL/. Anda dapat menanyakan tabel atau tampilan di database lain dengan SELECT pernyataan tersebut. Untuk informasi selengkapnya tentang menggunakan tautan database Oracle, lihat dokumentasi [Oracle](#).

Untuk informasi selengkapnya tentang menggunakan tautan database dengan AmazonRDS, lihat [RDSdokumentasi Amazon](#).

- Laporan AWS SCT penilaian menyediakan metrik server untuk konversi. Metrik tentang instans Oracle Anda ini mencakup yang berikut:
 - Komputasi dan kapasitas memori instans DB target.
 - Fitur Oracle yang tidak didukung seperti Real Application Clusters yang RDS tidak didukung Amazon.
 - Beban baca-tulis disk
 - Total throughput disk rata-rata
 - Informasi server seperti nama server, OS, nama host, dan set karakter.

Hak istimewa RDS untuk Oracle sebagai target

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

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



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

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

```
GRANT EXECUTE ON RDSADMIN.RDSADMIN_UTIL TO user_name;  
  
-- Server Level Objects (tablespace)  
GRANT CREATE TABLESPACE TO user_name;  
GRANT DROP TABLESPACE TO user_name;  
  
-- Server Level Objects (user roles)  
/* (grant source privileges with admin option or convert roles/privs as DBA) */  
  
-- Queues  
grant execute on DBMS_AQADM to user_name;  
grant aq_administrator_role to user_name;  
  
-- for Materialized View Logs creation  
GRANT SELECT ANY TABLE TO user_name;  
  
-- Roles  
GRANT RESOURCE TO user_name;  
GRANT CONNECT TO user_name;
```

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

Keterbatasan saat mengonversi Oracle ke Amazon RDS untuk Oracle

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

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

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

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

Menghubungkan ke SQL Database Postgre dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode database, dan kode aplikasi dari Postgre SQL ke target berikut:

- Amazon RDS untuk Saya SQL
- Amazon Aurora Edisi Kompatibel Saya SQL
- Amazon RDS para Postgre SQL
- Amazon Aurora SQL Postgre -Edisi Kompatibel

Untuk informasi selengkapnya, lihat bagian berikut:

Topik

- [Hak istimewa untuk Postgre SQL sebagai database sumber](#)
- [Menghubungkan ke Postgre SQL sebagai sumber](#)
- [Hak istimewa untuk My SQL sebagai basis data target](#)

Hak istimewa untuk Postgre SQL sebagai database sumber

Hak istimewa yang diperlukan untuk Postgre SQL sebagai sumber adalah sebagai berikut:

- CONNECTPADA DATABASE *<database_name>*
- USAGEPADA SCHEMA *<database_name>*
- SELECTDI ALL TABLES DALAM SCHEMA *<database_name>*
- SELECTDI ALL SEQUENCES DALAM SCHEMA *<database_name>*

Menghubungkan ke Postgre SQL sebagai sumber

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

Untuk terhubung ke database sumber Postgre SQL

1. Di dalam AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Postgre SQL, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

Parameter	Tindakan
Nama server	<p>Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.</p> <p>Anda dapat terhubung ke SQL database Postgre sumber Anda menggunakan protokol IPv6 alamat. Untuk melakukannya, pastikan Anda menggunakan tanda kurung siku untuk memasukkan alamat IP, seperti yang ditunjukkan pada contoh berikut.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; text-align: center; margin-top: 10px;">[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</div>

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

Parameter	Tindakan
Jalur pengemudi Postgre SQL	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

Hak istimewa untuk My SQL sebagai basis data target

Hak istimewa yang diperlukan untuk My SQL sebagai target saat Anda bermigrasi dari Postgre SQL adalah sebagai berikut:

- CREATEPADA * . *
- ALTERPADA * . *
- DROPPADA * . *
- INDEXPADA * . *
- REFERENCESPADA * . *
- SELECTPADA * . *
- CREATEVIEWPADA * . *
- SHOWVIEWPADA * . *
- TRIGGERPADA * . *
- CREATEROUTINEPADA * . *
- ALTERROUTINEPADA * . *
- EXECUTEPADA * . *
- INSERT, UPDATE PADA AWS _ POSTGRESQL _EXT. *
- INSERT,UPDATE, DELETE PADA AWS _ POSTGRESQL _ EXT _DATA. *

- `CREATETEMPORARYTABLESPADA AWS _ POSTGRESQL _ EXT _ DATA. *`

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

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

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

Untuk menggunakan Amazon RDS untuk My SQL atau Aurora My SQL sebagai target, tetapkan `lower_case_table_names` parameter ke 1. Nilai ini berarti bahwa SQL server Saya menangani pengidentifikasi nama objek seperti tabel, indeks, pemicu, dan database sebagai tidak peka huruf besar/kecil. Jika Anda telah mengaktifkan logging biner dalam instance target Anda, maka atur `log_bin_trust_function_creators` parameter ke 1. Dalam hal ini, Anda tidak perlu menggunakan `DETERMINISTIC`, `READS SQL DATA` atau `NO SQL` karakteristik untuk membuat fungsi yang disimpan. Untuk mengkonfigurasi parameter ini, buat grup parameter DB baru atau modifikasi grup parameter DB yang ada.

Menghubungkan ke SAP Database dengan AWS Schema Conversion Tool

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

- Amazon RDS untuk Saya SQL
- Amazon Aurora Edisi Kompatibel Saya SQL
- Amazon RDS untuk MariaDB
- Amazon RDS para Postgre SQL
- Amazon Aurora SQL Postgre -Edisi Kompatibel

Untuk informasi selengkapnya, lihat bagian berikut:

Topik

- [Hak istimewa untuk SAP ASE sebagai basis data sumber](#)
- [Menghubungkan ke SAP ASE \(Sybase\) sebagai sumber](#)
- [Hak istimewa untuk My SQL sebagai basis data target](#)
- [SAPASE ke pengaturan SQL konversi saya](#)
- [Hak istimewa untuk Postgre SQL sebagai basis data target](#)
- [SAPASE ke pengaturan konversi Postgre SQL](#)

Hak istimewa untuk SAP ASE sebagai basis data sumber

Untuk menggunakan SAP ASE database sebagai sumber, Anda membuat pengguna database dan memberikan izin. Untuk melakukan ini, ambil langkah-langkah berikut.

Membuat dan mengkonfigurasi pengguna database

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

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

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

```
USE <database_name>
sp_adduser min_privs
```

```
grant select on dbo.sysusers to min_privs
grant select on dbo.sysobjects to min_privs
grant select on dbo.sysindexes to min_privs
grant select on dbo.syscolumns to min_privs
grant select on dbo.sysreferences to min_privs
grant select on dbo.syscomments to min_privs
grant select on dbo.syspartitions to min_privs
grant select on dbo.syspartitionkeys to min_privs
grant select on dbo.sysconstraints to min_privs
grant select on dbo.systypes to min_privs
grant select on dbo.sysqueryplans to min_privs
```

Menghubungkan ke SAP ASE (Sybase) sebagai sumber

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

Untuk terhubung ke database SAP ASE sumber

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


Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

- Untuk memasukkan informasi koneksi basis data SAP ASE sumber secara manual, gunakan petunjuk berikut:

Parameter	Tindakan
Nama server	Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis Data	Masukkan nama SAP ASE database.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <div data-bbox="656 716 1507 1314" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px;"><p> Note</p><p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk database sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p></div>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada SSLtab:</p> <ul style="list-style-type: none">• Verifikasi sertifikat server: Pilih opsi ini untuk memverifikasi sertifikat server dengan menggunakan toko kepercayaan.• Toko kepercayaan: Lokasi toko kepercayaan yang berisi sertifikat.

Parameter	Tindakan
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Mengaktifkan opsi ini memungkinkan Anda menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.
SAPASEjalur pengemudi	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

Hak istimewa untuk My SQL sebagai basis data target

Keistimewaan yang diperlukan untuk My SQL sebagai target adalah sebagai berikut:

- CREATEPADA * . *
- ALTERPADA * . *
- DROPPADA * . *
- INDEXPADA * . *
- REFERENCESPADA * . *
- SELECTPADA * . *
- CREATEVIEWPADA * . *
- SHOWVIEWPADA * . *
- TRIGGERPADA * . *
- CREATEROUTINEPADA * . *

- ALTERROUTINEPADA *.*
- EXECUTE PADA *.*
- INSERT, UPDATE PADA AWS_SAPASE_EXT.*
- CREATETEMPORARYTABLESPADA AWS_SAPASE_EXT.*

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

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

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

Untuk menggunakan Amazon RDS untuk My SQL atau Aurora My SQL sebagai target, tetapkan `lower_case_table_names` parameter ke. 1 Nilai ini berarti bahwa SQL server Saya menangani pengidentifikasi nama objek seperti tabel, indeks, pemacu, dan database sebagai case insensitive. Jika Anda telah mengaktifkan logging biner dalam instance target Anda, maka atur `log_bin_trust_function_creators` parameter ke.1. Dalam hal ini, Anda tidak perlu menggunakan DETERMINISTIC, READS SQL DATA atau NO SQL karakteristik untuk membuat fungsi yang disimpan. Untuk mengkonfigurasi parameter ini, buat grup parameter DB baru atau modifikasi grup parameter DB yang ada.

SAPASE ke pengaturan SQL konversi saya

Untuk mengedit SAP ASE ke Pengaturan SQL konversi saya, pilih Pengaturan, lalu pilih Pengaturan konversi. Dari daftar atas, pilih SAPASE, lalu pilih SAPASE— Saya SQL atau SAP ASE — Amazon Aurora (SQL Kompatibel saya). AWS SCT menampilkan semua pengaturan yang tersedia SAP ASE untuk konversi PostgreSQL.

SAPASE ke Pengaturan SQL konversi saya AWS SCT termasuk opsi untuk yang berikut:

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

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

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

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

Secara default, AWS SCT mengkonversi nama-nama objek database, variabel, dan parameter ke huruf kecil. Untuk menyimpan kasus asli untuk nama-nama ini, pilih Perlakukan nama objek database sumber sebagai peka huruf besar/kecil. Pilih opsi ini jika Anda menggunakan nama objek peka huruf besar/kecil di server SAP ASE basis data sumber Anda.

Hak istimewa untuk Postgre SQL sebagai basis data target

Untuk menggunakan Postgre SQL sebagai target, AWS SCT membutuhkan hak istimewa. `CREATE ON DATABASE` Pastikan Anda memberikan hak istimewa ini untuk setiap basis data SQL Postgre target.

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

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

```
CREATE ROLE user_name LOGIN PASSWORD 'your_password';
```

```
GRANT CREATE ON DATABASE db_name TO user_name;  
ALTER DATABASE db_name SET SEARCH_PATH = "$user", public_synonyms, public;
```

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

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

Saat Anda menggunakan pengguna yang berbeda untuk mengonversi dan menerapkan skema yang berbeda ke basis data target Anda, Anda bisa mendapatkan pesan kesalahan saat tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan kesalahan ini, gunakan `superuser` peran.

SAPASE ke pengaturan konversi Postgre SQL

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

SAPASE ke pengaturan SQL konversi Postgre di AWS SCT menyertakan opsi untuk yang berikut:

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

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

Misalnya, untuk meminimalkan jumlah komentar dalam kode yang dikonversi, pilih Error only.

Untuk menyertakan komentar untuk semua item tindakan dalam kode yang dikonversi, pilih Semua pesan.

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

SQL

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

Secara default, AWS SCT mengkonversi nama-nama objek database, variabel, dan parameter ke huruf kecil. Untuk menyimpan kasus asli untuk nama-nama ini, pilih Perlakukan nama objek database sumber sebagai peka huruf besar/kecil. Pilih opsi ini jika Anda menggunakan nama objek peka huruf besar/kecil di server SAP ASE basis data sumber Anda.

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

- Untuk memungkinkan penggunaan indeks dengan nama yang sama di tabel yang berbeda di SAPASE.

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

Connect Microsoft SQL Server dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode database, dan kode aplikasi dari SQL Server ke target berikut:

- Amazon RDS untuk Saya SQL
- Amazon Aurora Edisi Kompatibel Saya SQL
- Amazon RDS para Postgre SQL
- Amazon Aurora SQL Postgre -Edisi Kompatibel
- Amazon RDS untuk SQL Server
- Amazon RDS untuk MariaDB

Note

AWS SCT tidak mendukung penggunaan Amazon RDS untuk SQL server sebagai sumber.

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

Topik

- [Hak istimewa untuk Microsoft SQL Server sebagai sumber](#)
- [Menggunakan Otentikasi Windows saat menggunakan Microsoft SQL Server sebagai sumber](#)
- [Menghubungkan ke SQL Server sebagai sumber](#)
- [Mengonversi SQL Server ke My SQL](#)
- [Migrasi dari SQL Server ke SQL Postgre dengan AWS Schema Conversion Tool](#)
- [Migrasi dari SQL Server ke Amazon RDS untuk SQL Server dengan AWS Schema Conversion Tool](#)

Hak istimewa untuk Microsoft SQL Server sebagai sumber

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

- VIEW DEFINITION
- VIEW DATABASE STATE

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

Ulangi hibah untuk setiap database yang skema Anda konversi.

Selain itu, berikan hak istimewa berikut pada master database:

- VIEW SERVER STATE
- VIEW ANY DEFINITION

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

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

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

Pada contoh sebelumnya, ganti *<user_name>* placeholder dengan nama pengguna yang Anda berikan dengan hak istimewa sebelumnya.

Untuk membaca informasi tentang Agen SQL Server, tambahkan pengguna Anda ke SQLAgentUser peran tersebut. Jalankan perintah berikut pada msdb database.

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

Pada contoh sebelumnya, ganti *<SQLAgentRole>* placeholder dengan nama peran Agen Server. SQL Kemudian ganti *<user_name>* placeholder dengan nama pengguna yang Anda berikan dengan hak istimewa sebelumnya. Untuk informasi selengkapnya, lihat [Menambahkan pengguna ke SQLAgentUser peran](#) di Panduan RDS Pengguna Amazon.

Untuk mendeteksi pengiriman log, berikan hak SELECT on dbo.log_shipping_primary_databases istimewa pada msdb database.

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

Menggunakan Otentikasi Windows saat menggunakan Microsoft SQL Server sebagai sumber

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

SQLServer memiliki dua mode akses: mode Otentikasi Windows dan Mode Campuran. Mode Otentikasi Windows memungkinkan Otentikasi Windows dan menonaktifkan SQL Otentikasi Server.

Mode Campuran memungkinkan Otentikasi Windows dan Otentikasi SQL Server. Otentikasi Windows selalu tersedia dan tidak dapat dinonaktifkan. Untuk informasi selengkapnya tentang Otentikasi Windows, lihat dokumentasi Microsoft Windows.

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

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

Menggunakan Otentikasi Windows dengan koneksi JDBC

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

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

Untuk menggunakan otentikasi terintegrasi

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

`sqljdbc_auth.dll` File diinstal di lokasi berikut:

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

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

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

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

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

- Salin sqljdbc_auth.dll ke folder tempatJDK/JREAnda berjalan. Anda dapat menyalin ke folder lib, folder bin, dan sebagainya. Sebagai contoh, Anda dapat menyalin ke folder berikut.

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

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

Note

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

Saat Anda terhubung ke database SQL Server, Anda dapat memilih Otentikasi Windows atau Otentikasi SQL Server untuk opsi Otentikasi.

Menghubungkan ke SQL Server sebagai sumber

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

Untuk menyambung ke database sumber Microsoft SQL Server

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

Parameter	Tindakan
Nama server	<p>Masukkan nama Layanan Nama Domain (DNS) atau alamat IP server basis data sumber Anda.</p> <p>Anda dapat terhubung ke database SQL Server sumber Anda menggunakan protokol IPv6 alamat. Untuk melakukannya, pastikan Anda menggunakan tanda kurung siku untuk memasukkan alamat IP, seperti yang ditunjukkan pada contoh berikut.</p> <div style="border: 1px solid #ccc; border-radius: 10px; padding: 5px; width: fit-content; margin: 10px auto;"> <p>[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]</p> </div>
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Nama instans	Masukkan nama instance untuk database SQL Server. Untuk menemukan nama instance, jalankan kueri <code>SELECT @@servername;</code> pada database SQL Server Anda.
Autentikasi	Pilih jenis otentikasi dari Windows Authentication dan SQLServer Authentication.

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

Parameter	Tindakan
Jalur Pengemudi Server Sql	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>
Pustaka Otentikasi Windows	<p>Masukkan path ke sqljdbc_auth.dll file. Secara default, file ini diinstal di lokasi berikut:</p> <p><i><installation directory of the JDBC driver>sqljdbc_<version> \<language> \auth\</i></p>

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

Mengonversi SQL Server ke My SQL

Untuk meniru fungsi database Microsoft SQL Server dalam SQL kode Saya yang dikonversi, gunakan paket SQL ekstensi SQL Server ke Saya. AWS SCT Untuk informasi selengkapnya tentang paket ekstensi, lihat [Menggunakan paket ekstensi dengan AWS Schema Conversion Tool](#).

Topik

- [Hak istimewa untuk My SQL sebagai basis data target](#)
- [SQLServer ke pengaturan SQL konversi saya](#)
- [Pertimbangan migrasi](#)

Hak istimewa untuk My SQL sebagai basis data target

Keistimewaan yang diperlukan untuk My SQL sebagai target adalah sebagai berikut:

- CREATEPADA *.*
- ALTERPADA *.*
- DROPPADA *.*
- INDEXPADA *.*
- REFERENCESPADA *.*
- SELECTPADA *.*
- CREATEVIEWPADA *.*
- SHOWVIEWPADA *.*
- TRIGGERPADA *.*
- CREATEROUTINEPADA *.*
- ALTERROUTINEPADA *.*
- EXECUTEPADA *.*
- INSERT, UPDATE PADA AWS _ SQLSERVER _ EXT. *
- INSERT, UPDATE, DELETE PADA AWS _ SQLSERVER _ EXT _ DATA. *
- CREATETEMPORARYTABLESPADA AWS _ SQLSERVER _ EXT _ DATA. *

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

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


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

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

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

Untuk menggunakan Amazon RDS untuk My SQL atau Aurora My SQL sebagai target, tetapkan `lower_case_table_names` parameternya ke 1. Nilai ini berarti bahwa SQL server Saya menangani pengidentifikasi nama objek seperti tabel, indeks, pemacu, dan database sebagai tidak peka huruf besar/kecil. Jika Anda telah mengaktifkan logging biner dalam instance target Anda, maka atur `log_bin_trust_function_creators` parameternya ke 1. Dalam hal ini, Anda tidak perlu menggunakan `DETERMINISTIC`, `READS SQL DATA` atau `NO SQL` karakteristik untuk membuat fungsi yang disimpan. Untuk mengkonfigurasi parameter ini, buat grup parameter DB baru atau modifikasi grup parameter DB yang ada.

SQLServer ke pengaturan SQL konversi saya

Untuk mengedit SQL Server ke pengaturan SQL konversi saya, AWS SCT pilih Pengaturan, lalu pilih Pengaturan konversi. Dari daftar atas, pilih SQLServer, lalu pilih SQLServer — Saya SQL. AWS SCT menampilkan semua pengaturan yang tersedia untuk SQL Server ke SQL konversi Saya.

SQLServer ke pengaturan SQL konversi Saya AWS SCT termasuk opsi untuk hal berikut:

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

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

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

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

Pertimbangan migrasi

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

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

Untuk menggunakan emulasi yang benarINSERT ON DUPLICATE KEY, pastikan bahwa ada kendala unik atau kunci utama pada target database Saya. SQL

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

Saya SQL tidak menggunakan GOTO pernyataan. Ketika AWS SCT mengkonversi kode yang berisi GOTO pernyataan, itu mengubah pernyataan untuk menggunakan pernyataan BEGIN...END atauLOOP...END LOOP. Anda dapat menemukan contoh bagaimana AWS SCT mengkonversi GOTO pernyataan dalam tabel berikut.

SQLPernyataan server	SQLPernyataan saya
<pre> BEGIN statement1; GOTO label1; statement2; label1: Statement3; END </pre>	<pre> BEGIN label1: BEGIN statement1; LEAVE label1; statement2; END; Statement3; END </pre>
<pre> BEGIN </pre>	<pre> BEGIN </pre>

SQLPernyataan server	SQLPernyataan saya
<pre> statement1; label1: statement2; GOTO label1; statement3; statement4; END </pre>	<pre> statement1; label1: LOOP statement2; ITERATE label1; LEAVE label1; END LOOP; statement3; statement4; END </pre>
<pre> BEGIN statement1; label1: statement2; statement3; statement4; END </pre>	<pre> BEGIN statement1; label1: BEGIN statement2; statement3; statement4; END; END </pre>

- Saya SQL tidak mendukung fungsi bernilai tabel multistatement. AWS SCT mensimulasikan fungsi bernilai tabel selama konversi dengan membuat tabel sementara dan menulis ulang pernyataan untuk menggunakan tabel sementara ini.

Migrasi dari SQL Server ke SQL Postgre dengan AWS Schema Conversion Tool

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

Topik

- [Hak istimewa untuk Postgre SQL sebagai basis data target](#)
- [SQLPengaturan konversi server ke Postgre SQL](#)
- [Mengonversi partisi SQL Server ke partisi Postgre SQL versi 10](#)
- [Pertimbangan migrasi](#)
- [Menggunakan paket AWS SCT ekstensi untuk meniru Agen SQL Server di Postgre SQL](#)
- [Menggunakan paket AWS SCT ekstensi untuk meniru SQL Server Database Mail di Postgre SQL](#)

Hak istimewa untuk Postgre SQL sebagai basis data target

Untuk menggunakan Postgre SQL sebagai target, AWS SCT membutuhkan hak istimewa. CREATE ON DATABASE Pastikan Anda memberikan hak istimewa ini untuk setiap basis data SQL Postgre target.

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

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

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

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

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

Saat Anda menggunakan pengguna yang berbeda untuk mengonversi dan menerapkan skema yang berbeda ke basis data target Anda, Anda bisa mendapatkan pesan kesalahan saat tidak AWS SCT dapat menjatuhkan skema. Untuk menghindari pesan kesalahan ini, gunakan `superuser` peran.

SQLPengaturan konversi server ke Postgre SQL

Untuk mengedit pengaturan SQL konversi SQL Server ke Postgre, pilih Pengaturan, lalu pilih Pengaturan konversi. Dari daftar atas, pilih SQLServer, lalu pilih SQLServer — Postgre SQL. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi SQL Server ke PostgreSQL.

SQLPengaturan SQL konversi server ke Postgre di AWS SCT menyertakan opsi untuk yang berikut:

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

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

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

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

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

- Untuk mengonversi prosedur SQL Server ke fungsi PostgreSQL.

Postgre SQL versi 10 dan sebelumnya tidak mendukung prosedur. Untuk pelanggan yang tidak terbiasa menggunakan prosedur di PostgreSQL, AWS SCT dapat mengonversi prosedur ke fungsi. Untuk melakukannya, pilih Konversi prosedur ke fungsi.

- Untuk meniru output EXEC dalam tabel.

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

- Untuk menentukan template yang akan digunakan untuk nama skema dalam kode yang dikonversi. Untuk templat pembuatan nama Skema, pilih salah satu opsi berikut:
 - `<source_db>`— Menggunakan nama database SQL Server sebagai nama skema di SQL Postgre.
 - `<source_schema>`— Menggunakan nama skema SQL Server sebagai nama skema di Postgre. SQL
 - `_ <source_db><schema>`— Menggunakan kombinasi database SQL Server dan nama skema sebagai nama skema di Postgre. SQL
- Untuk menyimpan huruf huruf nama objek sumber Anda.

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

- Untuk menyimpan nama parameter dari database sumber Anda.

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

Mengonversi partisi SQL Server ke partisi Postgre SQL versi 10

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

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

- SQLServer memungkinkan Anda untuk partisi tabel menggunakan kolom tanpa NOT NULL kendala. Dalam hal ini, semua NULL nilai masuk ke partisi paling kiri. Postgre SQL tidak mendukung NULL nilai untuk RANGE partisi.
- SQLServer memungkinkan Anda membuat kunci primer dan unik untuk tabel yang dipartisi. Untuk PostgreSQL, Anda membuat kunci primer atau unik untuk setiap partisi secara langsung. Jadi, PRIMARY atau UNIQUE KEY kendala harus dihapus dari tabel

induknya saat bermigrasi ke Postgre. SQL Nama kunci yang dihasilkan mengambil format `<original_key_name>_<partition_number>`.

- SQLServer memungkinkan Anda membuat batasan kunci asing dari dan ke tabel yang dipartisi. Postgre SQL tidak mendukung kunci asing yang mereferensikan tabel yang dipartisi. Juga, Postgre SQL tidak mendukung referensi kunci asing dari tabel yang dipartisi ke tabel lain.
- SQLServer memungkinkan Anda membuat indeks untuk tabel yang dipartisi. Untuk PostgreSQL, indeks harus dibuat untuk setiap partisi secara langsung. Dengan demikian, indeks harus dihapus dari tabel induknya saat bermigrasi ke Postgre. SQL Nama indeks yang dihasilkan mengambil format `<original_index_name>_<partition_number>`.
- Postgre SQL tidak mendukung indeks yang dipartisi.

Pertimbangan migrasi

Beberapa hal yang perlu dipertimbangkan saat memigrasi skema SQL Server ke Postgre: SQL

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

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

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

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

SQLGOTOPernyataan server dan pernyataan SQL Postgre yang dikonversi

SQLPernyataan server	Pernyataan Postgre SQL
<pre>BEGIN statement1; GOTO label1; statement2; label1: Statement3; END</pre>	<pre>BEGIN label1: BEGIN statement1; EXIT label1; statement2; 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>

SQL Pernyataan server

```

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

```

Pernyataan Postgre SQL

```

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

```

- Postgre SQL tidak mendukung pernyataan. MERGE AWS SCT mengemulasi perilaku MERGE pernyataan dengan cara-cara berikut:
 - Dengan CONFLICT konstruksi INSERT ON.
 - Dengan menggunakan UPDATE FROM DML pernyataan, seperti MERGE tanpa WHEN NOT MATCHED klausa.
 - Dengan menggunakan CURSOR, seperti dengan DELETE klausa MERGE with atau dengan menggunakan pernyataan kondisi MERGE ON yang kompleks.
- AWS SCT dapat menambahkan pemicu database ke pohon objek saat Amazon RDS adalah targetnya.
- AWS SCT dapat menambahkan pemicu tingkat server ke pohon objek saat Amazon RDS adalah targetnya.
- SQL Server secara otomatis membuat dan mengelola deleted dan inserted tabel. Anda dapat menggunakan tabel sementara, memori-residen ini untuk menguji efek modifikasi data tertentu dan untuk mengatur kondisi untuk tindakan DML pemicu. AWS SCT dapat mengonversi penggunaan tabel ini di dalam pernyataan DML pemicu.
- AWS SCT dapat menambahkan server tertaut ke pohon objek saat Amazon RDS adalah targetnya.
- Saat bermigrasi dari Microsoft SQL Server ke PostgreSQL, SNAME fungsi SUSER _ bawaan dikonversi sebagai berikut:

- `SUSER_ SNAME` — Mengembalikan nama login yang terkait dengan nomor identifikasi keamanan (SID).
- `SUSER_ SNAME (<server_user_sid>)` - Tidak didukung.
- `SUSER_ SNAME () CURRENT _ USER` — Mengembalikan nama pengguna dari konteks eksekusi saat ini.
- `SUSER_ SNAME (NULL)` - PengembalianNULL.
- Mengonversi fungsi bernilai tabel didukung. Fungsi bernilai tabel mengembalikan tabel dan dapat menggantikan tabel dalam kueri.
- `PATINDEX` mengembalikan posisi awal dari kemunculan pertama pola dalam ekspresi tertentu pada semua teks yang valid dan tipe data karakter. Ia mengembalikan nol jika pola tidak ditemukan. `<pattern character><expression character varying>` Saat mengonversi dari SQL Server ke Amazon RDS untuk PostgreSQL, ganti kode aplikasi yang digunakan `PATINDEX` dengan AWS SCT `aws_sqlserver_ext.patindex (,)`.
- Di SQL Server, tipe tabel yang ditentukan pengguna adalah tipe yang mewakili definisi struktur tabel. Anda menggunakan tipe tabel yang ditentukan pengguna untuk mendeklarasikan parameter nilai tabel untuk prosedur atau fungsi yang disimpan. Anda juga dapat menggunakan tipe tabel yang ditentukan pengguna untuk mendeklarasikan variabel tabel yang ingin Anda gunakan dalam batch atau dalam badan prosedur atau fungsi yang disimpan. AWS SCT meniru jenis ini di PostgreSQL dengan membuat tabel sementara.

Saat mengonversi dari SQL Server ke PostgreSQL, AWS SCT mengubah objek sistem SQL Server menjadi objek yang dapat dikenali di Postgre. SQL Tabel berikut menunjukkan bagaimana objek sistem dikonversi.

Kasus penggunaan MS SQL Server	Substitusi Postgre SQL
<code>SYS.SCHEMAS</code>	<code>AWS_SQLSERVER_EXT.SYS_SCHEMAS</code>
<code>SYS.TABLES</code>	<code>AWS_SQLSERVER_EXT.SYS_TABLES</code>
<code>SYS.VIEWS</code>	<code>AWS_SQLSERVER_EXT.SYS_VIEWS</code>
<code>SYS.ALL_VIEWS</code>	<code>AWS_SQLSERVER_EXT.SYS_ALL_VIEWS</code>

Kasus penggunaan MS SQL Server	Substitusi Postgre SQL
SYS.TYPES	AWS_SQLSERVER_EXT.SYS_TYPES
SYS.COLUMNS	AWS_SQLSERVER_EXT.SYS_COLUMNS
SYS.ALL_COLUMNS	AWS_SQLSERVER_EXT.SYS_ALL_COLUMNS
SYS.FOREIGN_KEYS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEYS
SYS.SYSFOREIGNKEYS	AWS_SQLSERVER_EXT.SYS_SYSFOREIGNKEYS
SYS.FOREIGN_KEY_COLUMNS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEY_COLUMNS
SYS.KEY_CONSTRAINTS	AWS_SQLSERVER_EXT.SYS_KEY_CONSTRAINTS
SYS.IDENTITY_COLUMNS	AWS_SQLSERVER_EXT.SYS_IDENTITY_COLUMNS
SYS.PROCEDURES	AWS_SQLSERVER_EXT.SYS_PROCEDURES
SYS.INDEXES	AWS_SQLSERVER_EXT.SYS_INDEXES
SYS.SYSINDEXES	AWS_SQLSERVER_EXT.SYS_SYSINDEXES
SYS.OBJECTS	AWS_SQLSERVER_EXT.SYS_OBJECTS
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

Kasus penggunaan MS SQL Server	Substitusi Postgre SQL
INFORMATION_SCHEMA.VIEWS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_VIEWS
INFORMATION_SCHEMA.TABLES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLES
INFORMATION_SCHEMA.COLUMNS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_COLUMNS
INFORMATION_SCHEMA.CHECK_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CHECK_CONSTRAINTS
INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_REFERENTIAL_CONSTRAINTS
INFORMATION_SCHEMA.TABLE_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLE_CONSTRAINTS
INFORMATION_SCHEMA.KEY_COLUMN_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_KEY_COLUMN_USAGE
INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAINT_TABLE_USAGE
INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CONSTRAINT_COLUMN_USAGE
INFORMATION_SCHEMA.ROUTINES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_ROUTINES
SYS.SYSPROCESSES	AWS_SQLSERVER_EXT.SYS_SYSPROCESSES
sys.system_objects	AWS_SQLSERVER_EXT.SYS_SYSTEM_OBJECTS

Menggunakan paket AWS SCT ekstensi untuk meniru Agen SQL Server di Postgre SQL

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

Postgre SQL tidak memiliki padanan untuk SQL Server Agent. Untuk meniru fitur Agen SQL Server, AWS SCT buat paket ekstensi. Paket ekstensi ini menggunakan AWS Lambda dan Amazon CloudWatch. AWS Lambda mengimplementasikan antarmuka yang Anda gunakan untuk mengelola jadwal dan menjalankan pekerjaan. Amazon CloudWatch mempertahankan aturan jadwal.

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

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

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

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

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

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

```
}  
}
```

Untuk membuat pekerjaan dengan beberapa langkah, gunakan format berikut.

```
{  
  "mode": "run_job",  
  "parameters": {  
    "job_name": "Job1",  
    "enabled": "true",  
    "start_step_id": 1,  
    "notify_level_email": [0|1|2|3],  
    "notify_email": email,  
    "delete_level": [0|1|2|3],  
    "job_callback": "ProcCallBackJob(job_name, code, message)",  
    "step_callback": "ProcCallBackStep(job_name, step_id, code, message)"  
  },  
  "steps": [  
    {  
      "id":1,  
      "cmd": "ProcStep1",  
      "cmdexec_success_code": 0,  
      "on_success_action": [2|3|4],  
      "on_success_step_id": 1,  
      "on_fail_action": 0,  
      "on_fail_step_id": 0,  
      "retry_attempts": number,  
      "retry_interval": number  
    },  
    {  
      "id":2,  
      "cmd": "ProcStep2",  
      "cmdexec_success_code": 0,  
      "on_success_action": [1|2|3|4],  
      "on_success_step_id": 0,  
      "on_fail_action": 0,  
      "on_fail_step_id": 0,  
      "retry_attempts": number,  
      "retry_interval": number  
    },  
    ...  
  ]  
}
```

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

Tabel yang meniru Agen SQL Server di Postgre SQL

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

sysjobs

Menyimpan informasi tentang pekerjaan.

sysjobsteps

Menyimpan informasi tentang langkah-langkah pekerjaan.

sysschedules

Menyimpan informasi tentang jadwal pekerjaan.

jadwal sysjob

Menyimpan informasi jadwal untuk pekerjaan individu.

sysjobhistory

Menyimpan informasi tentang menjalankan pekerjaan terjadwal.

Prosedur yang meniru SQL Server Agent di Postgre SQL

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

sp_add_pekerjaan

Menambahkan pekerjaan baru.

sp_add_jobstep

Menambahkan langkah ke pekerjaan.

sp_add_schedule

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

sp_attach_schedule

Menetapkan jadwal untuk pekerjaan yang dipilih.

sp_add_jobschedule

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

sp_update_job

Memperbarui atribut pekerjaan yang dibuat sebelumnya.

sp_update_jobstep

Memperbarui atribut langkah dalam pekerjaan.

sp_update_schedule

Memperbarui atribut aturan jadwal di Amazon CloudWatch.

sp_update_jobschedule

Memperbarui atribut jadwal untuk pekerjaan yang ditentukan.

sp_delete_job

Menghapus pekerjaan.

sp_delete_jobstep

Menghapus langkah pekerjaan dari pekerjaan.

sp_delete_schedule

Menghapus jadwal.

sp_delete_jobschedule

Menghapus aturan jadwal untuk pekerjaan yang ditentukan dari Amazon CloudWatch.

sp_detach_schedule

Menghapus hubungan antara jadwal dan pekerjaan.

get_jobs, update_job

Prosedur internal yang berinteraksi dengan AWS Elastic Beanstalk.

sp_verify_job_date, sp_verify_job_time, sp_verify_job, sp_verify_jobstep, sp_verify_job_identifiers,
sp_verify_schedule_identifiers

Prosedur internal yang memeriksa pengaturan.

Sintaks untuk prosedur yang meniru Agen SQL Server di Postgre SQL

`aws_sqlserver_ext.sp_add_job` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_add_job` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_job_name varchar,  
par_enabled smallint = 1,  
par_description varchar = NULL::character varying,  
par_start_step_id integer = 1,  
par_category_name varchar = NULL::character varying,  
par_category_id integer = NULL::integer,  
par_owner_login_name varchar = NULL::character varying,  
par_notify_level_eventlog integer = 2,  
par_notify_level_email integer = 0,  
par_notify_level_netsend integer = 0,  
par_notify_level_page integer = 0,  
par_notify_email_operator_name varchar = NULL::character varying,  
par_notify_netsend_operator_name varchar = NULL::character varying,  
par_notify_page_operator_name varchar = NULL::character varying,  
par_delete_level integer = 0,  
inout par_job_id integer = NULL::integer,  
par_originating_server varchar = NULL::character varying,  
out returncode integer
```

`aws_sqlserver_ext.sp_add_jobstep` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_add_jobstep` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_step_id integer = NULL::integer,  
par_step_name varchar = NULL::character varying,  
par_subsystem varchar = 'TSQL'::bpchar,  
par_command text = NULL::text,  
par_additional_parameters text = NULL::text,  
par_cmdexec_success_code integer = 0,  
par_on_success_action smallint = 1,  
par_on_success_step_id integer = 0,  
par_on_fail_action smallint = 2,  
par_on_fail_step_id integer = 0,  
par_server varchar = NULL::character varying,
```

```
par_database_name varchar = NULL::character varying,  
par_database_user_name varchar = NULL::character varying,  
par_retry_attempts integer = 0,  
par_retry_interval integer = 0,  
par_os_run_priority integer = 0,  
par_output_file_name varchar = NULL::character varying,  
par_flags integer = 0,  
par_proxy_id integer = NULL::integer,  
par_proxy_name varchar = NULL::character varying,  
inout par_step_uid char = NULL::bpchar,  
out returncode integer
```

`aws_sqlserver_ext.sp_add_schedule` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_add_schedule` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_schedule_name varchar,  
par_enabled smallint = 1,  
par_freq_type integer = 0,  
par_freq_interval integer = 0,  
par_freq_subday_type integer = 0,  
par_freq_subday_interval integer = 0,  
par_freq_relative_interval integer = 0,  
par_freq_recurrence_factor integer = 0,  
par_active_start_date integer = NULL::integer,  
par_active_end_date integer = 99991231,  
par_active_start_time integer = 0,  
par_active_end_time integer = 235959,  
par_owner_login_name varchar = NULL::character varying,  
*inout par_schedule_uid char = NULL::bpchar,*  
inout par_schedule_id integer = NULL::integer,  
par_originating_server varchar = NULL::character varying,  
out returncode integer
```

`aws_sqlserver_ext.sp_attach_schedule` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_attach_schedule` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_schedule_id integer = NULL::integer,  
par_schedule_name varchar = NULL::character varying,
```

```
par_automatic_post smallint = 1,  
out returncode integer
```

`aws_sqlserver_ext.sp_add_jobschedule` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_add_jobschedule` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_name varchar = NULL::character varying,  
par_enabled smallint = 1,  
par_freq_type integer = 1,  
par_freq_interval integer = 0,  
par_freq_subday_type integer = 0,  
par_freq_subday_interval integer = 0,  
par_freq_relative_interval integer = 0,  
par_freq_recurrence_factor integer = 0,  
par_active_start_date integer = NULL::integer,  
par_active_end_date integer = 99991231,  
par_active_start_time integer = 0,  
par_active_end_time integer = 235959,  
inout par_schedule_id integer = NULL::integer,  
par_automatic_post smallint = 1,  
inout par_schedule_uid char = NULL::bpchar,  
out returncode integer
```

`aws_sqlserver_ext.sp_delete_job` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_delete_job` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_originating_server varchar = NULL::character varying,  
par_delete_history smallint = 1,  
par_delete_unused_schedule smallint = 1,  
out returncode integer
```

`aws_sqlserver_ext.sp_delete_jobstep` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_delete_jobstep` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

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

`aws_sqlserver_ext.sp_delete_jobschedule` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_delete_jobschedule` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_name varchar = NULL::character varying,  
par_keep_schedule integer = 0,  
par_automatic_post smallint = 1,  
out_returncode integer
```

`aws_sqlserver_ext.sp_delete_schedule` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_delete_schedule` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_schedule_id integer = NULL::integer,  
par_schedule_name varchar = NULL::character varying,  
par_force_delete smallint = 0,  
par_automatic_post smallint = 1,  
out_returncode integer
```

`aws_sqlserver_ext.sp_detach_schedule` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_detach_schedule` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_job_id integer = NULL::integer,  
par_job_name varchar = NULL::character varying,  
par_schedule_id integer = NULL::integer,  
par_schedule_name varchar = NULL::character varying,  
par_delete_unused_schedule smallint = 0,  
par_automatic_post smallint = 1,  
out_returncode integer
```

`aws_sqlserver_ext.sp_update_job` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_update_job` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_job_id integer = NULL::integer
par_job_name varchar = NULL::character varying
par_new_name varchar = NULL::character varying
par_enabled smallint = NULL::smallint
par_description varchar = NULL::character varying
par_start_step_id integer = NULL::integer
par_category_name varchar = NULL::character varying
par_owner_login_name varchar = NULL::character varying
par_notify_level_eventlog integer = NULL::integer
par_notify_level_email integer = NULL::integer
par_notify_level_netsend integer = NULL::integer
par_notify_level_page integer = NULL::integer
par_notify_email_operator_name varchar = NULL::character varying
par_notify_netsend_operator_name varchar = NULL::character varying
par_notify_page_operator_name varchar = NULL::character varying
par_delete_level integer = NULL::integer
par_automatic_post smallint = 1
out returncode integer
```

`aws_sqlserver_ext.sp_update_jobschedule` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_update_jobschedule` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_job_id integer = NULL::integer
par_job_name varchar = NULL::character varying
par_name varchar = NULL::character varying
par_new_name varchar = NULL::character varying
par_enabled smallint = NULL::smallint
par_freq_type integer = NULL::integer
par_freq_interval integer = NULL::integer
par_freq_subday_type integer = NULL::integer
par_freq_subday_interval integer = NULL::integer
par_freq_relative_interval integer = NULL::integer
par_freq_recurrence_factor integer = NULL::integer
par_active_start_date integer = NULL::integer
par_active_end_date integer = NULL::integer
par_active_start_time integer = NULL::integer
par_active_end_time integer = NULL::integer
```

```
par_automatic_post smallint = 1
out_returncode integer
```

`aws_sqlserver_ext.sp_update_jobstep` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_update_jobstep` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_job_id integer = NULL::integer
par_job_name varchar = NULL::character varying
par_step_id integer = NULL::integer
par_step_name varchar = NULL::character varying
par_subsystem varchar = NULL::character varying
par_command text = NULL::text
par_additional_parameters text = NULL::text
par_cmdexec_success_code integer = NULL::integer
par_on_success_action smallint = NULL::smallint
par_on_success_step_id integer = NULL::integer
par_on_fail_action smallint = NULL::smallint
par_on_fail_step_id integer = NULL::integer
par_server varchar = NULL::character varying
par_database_name varchar = NULL::character varying
par_database_user_name varchar = NULL::character varying
par_retry_attempts integer = NULL::integer
par_retry_interval integer = NULL::integer
par_os_run_priority integer = NULL::integer
par_output_file_name varchar = NULL::character varying
par_flags integer = NULL::integer
par_proxy_id integer = NULL::integer
par_proxy_name varchar = NULL::character varying
out_returncode integer
```

`aws_sqlserver_ext.sp_update_schedule` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_update_schedule` prosedur. Untuk informasi selengkapnya tentang prosedur Agen SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_schedule_id integer = NULL::integer
par_name varchar = NULL::character varying
par_new_name varchar = NULL::character varying
par_enabled smallint = NULL::smallint
par_freq_type integer = NULL::integer
par_freq_interval integer = NULL::integer
par_freq_subday_type integer = NULL::integer
```

```
par_freq_subday_interval integer = NULL::integer
par_freq_relative_interval integer = NULL::integer
par_freq_recurrence_factor integer = NULL::integer
par_active_start_date integer = NULL::integer
par_active_end_date integer = NULL::integer
par_active_start_time integer = NULL::integer
par_active_end_time integer = NULL::integer
par_owner_login_name varchar = NULL::character varying
par_automatic_post smallint = 1
out returncode integer
```

Contoh untuk menggunakan prosedur yang meniru Agen SQL Server di Postgre SQL

Untuk menambahkan pekerjaan baru, gunakan `aws_sqlserver_ext.sp_add_job` prosedur seperti yang ditunjukkan berikut.

```
SELECT * FROM aws_sqlserver_ext.sp_add_job (
    par_job_name := 'test_job',
    par_enabled := 1::smallint,
    par_start_step_id := 1::integer,
    par_category_name := '[Uncategorized (Local)]',
    par_owner_login_name := 'sa');
```

Untuk menambahkan langkah pekerjaan baru, gunakan `aws_sqlserver_ext.sp_add_jobstep` prosedur seperti yang ditunjukkan berikut.

```
SELECT * FROM aws_sqlserver_ext.sp_add_jobstep (
    par_job_name := 'test_job',
    par_step_id := 1::smallint,
    par_step_name := 'test_job_step1',
    par_subsystem := 'TSQL',
    par_command := 'EXECUTE [dbo].[PROC_TEST_JOB_STEP1];',
    par_server := NULL,
    par_database_name := 'GOLD_TEST_SS');
```

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

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

```
par_active_start_time := 233000);
```

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

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

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

```
SELECT * FROM aws_sqlserver_ext.sp_add_jobschedule (  
    par_job_name := 'test_job2',  
    par_name := 'test_schedule2',  
    par_enabled := 1::smallint,  
    par_freq_type := 4,  
    par_freq_interval := 1,  
    par_freq_subday_type := 4,  
    par_freq_subday_interval := 1,  
    par_freq_relative_interval := 0,  
    par_freq_recurrence_factor := 0,  
    par_active_start_date := 20100801,  
    par_active_end_date := 99991231,  
    par_active_start_time := 0,  
    par_active_end_time := 0);
```

Gunakan contoh kasus untuk meniru Agen SQL Server di Postgre SQL

Jika kode basis data sumber Anda menggunakan Agen SQL Server untuk menjalankan pekerjaan, Anda dapat menggunakan paket SQL ekstensi SQL Server ke Postgre untuk mengonversi kode ini AWS SCT ke Postgre. SQL Paket ekstensi menggunakan AWS Lambda fungsi untuk meniru perilaku Agen SQL Server.

Anda dapat membuat AWS Lambda fungsi baru atau mendaftarkan fungsi yang ada.

Untuk membuat AWS Lambda fungsi baru

1. Di AWS SCT, di pohon database target, buka menu konteks (klik kanan), pilih Terapkan paket ekstensi untuk, lalu pilih SQLPostgre.

Wizard paket ekstensi muncul.

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

- Pilih Buat AWS Lambda fungsi.
- Untuk login Database, masukkan nama pengguna database target.
- Untuk kata sandi Database, masukkan kata sandi untuk nama pengguna yang Anda masukkan pada langkah sebelumnya.
- Untuk folder pustaka Python, masukkan path ke folder pustaka Python Anda.
- Pilih Create AWS Lambda Function, lalu pilih Next.

Untuk mendaftarkan AWS Lambda fungsi yang Anda gunakan sebelumnya

- Jalankan skrip berikut pada database target Anda.

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

Dalam contoh sebelumnya, *ARN* adalah Amazon Resource Name (ARN) dari AWS Lambda fungsi yang diterapkan.

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

Untuk membuat tugas sederhana ini

1. Buat pekerjaan menggunakan `aws_sqlserver_ext.sp_add_job` fungsi seperti yang ditunjukkan berikut.

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

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

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

Langkah pekerjaan menentukan apa fungsi yang dilakukan.

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

```
SELECT
FROM aws_sqlserver_ext.sp_add_jobschedule (
    par_job_name := 'test_simple_job',
    par_name := 'test_schedule',
    par_freq_type := 4, /* Daily */
    par_freq_interval := 1, /* frequency_interval is unused */
    par_freq_subday_type := 4, /* Minutes */
    par_freq_subday_interval := 5 /* 5 minutes */);
```

Langkah pekerjaan menentukan apa fungsi yang dilakukan.

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

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

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

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

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

Untuk akun email baru, gunakan perintah berikut.

```
do
$$
begin
PERFORM sysmail_add_account_sp (
    par_account_name := 'your_account_name',
    par_email_address := 'your_account_email',
    par_display_name := 'your_account_display_name',
    par_mailserver_type := 'AWSLAMBDA'
    par_mailserver_name := 'ARN'
);
end;
$$ language plpgsql;
```

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

```
do
$$
begin
PERFORM sysmail_update_account_sp (
    par_account_name := 'existind_account_name',
    par_mailserver_type := 'AWSLAMBDA'
    par_mailserver_name := 'ARN'
);
end;
$$ language plpgsql;
```

Dalam contoh sebelumnya, *ARN* adalah fungsi ARN Lambda Anda.

Untuk meniru perilaku SQL Server Database Mail di PostgreSQL, paket AWS SCT ekstensi menggunakan tabel, tampilan, dan prosedur berikut.

Tabel yang meniru SQL Server Database Mail di Postgre SQL

Untuk meniru SQL Server Database Mail, paket ekstensi menggunakan tabel berikut:

sysmail_akun

Menyimpan informasi tentang akun email.

sysmail_profil

Menyimpan informasi tentang profil pengguna.

sysmail_server

Menyimpan informasi tentang server email.

sysmail_mailitems

Menyimpan daftar pesan email.

lampiran sysmail_

Berisi satu baris untuk setiap lampiran email.

sysmail_log

Menyimpan informasi layanan tentang mengirim pesan email.

sysmail_profileakun

Menyimpan informasi tentang profil pengguna dan akun email.

Tampilan yang meniru SQL Server Database Mail di Postgre SQL

Untuk meniru SQL Server Database Mail, AWS SCT membuat tampilan berikut dalam database Postgre untuk memastikan SQL kompatibilitas. Paket ekstensi tidak menggunakannya, tetapi kode yang dikonversi dapat menanyakan tampilan ini.

sysmail_allitems

Termasuk daftar semua email.

sysmail_faileditems

Termasuk daftar email yang tidak dapat dikirim.

sysmail_sentitems

Termasuk daftar email yang dikirim.

sysmail_unsentitems

Termasuk daftar email yang belum terkirim.

sysmail_mailattachments

Termasuk daftar file terlampir.

Prosedur yang meniru SQL Server Database Mail di Postgre SQL

Untuk meniru SQL Server Database Mail, paket ekstensi menggunakan prosedur berikut:

sp_send_dbmail

Mengirim email ke penerima yang ditentukan.

sysmail_add_profile_sp

Membuat profil pengguna baru.

sysmail_add_account_sp

Membuat akun email baru yang menyimpan informasi seperti kredensi Simple Mail Transfer Protocol (SMTP), dan sebagainya.

sysmail_add_profileaccount_sp

Menambahkan akun email ke profil pengguna yang ditentukan.

sysmail_update_profile_sp

Mengubah atribut profil pengguna seperti deskripsi, nama, dan sebagainya.

sysmail_update_account_sp

Mengubah informasi di akun email yang ada.

sysmail_update_profileaccount_sp

Memperbarui informasi akun email di profil pengguna yang ditentukan.

sysmail_delete_profileaccount_sp

Menghapus akun email dari profil pengguna yang ditentukan.

sysmail_delete_account_sp

Menghapus akun email.

sysmail_delete_profile_sp

Menghapus profil pengguna.

sysmail_delete_mailitems_sp

Menghapus email dari tabel internal.

sysmail_help_profile_sp

Menampilkan informasi tentang profil pengguna.

sysmail_help_account_sp

Menampilkan informasi tentang akun email.

sysmail_help_profileaccount_sp

Menampilkan informasi tentang akun email yang terkait dengan profil pengguna.

sysmail_dbmail_json

Prosedur internal yang menghasilkan JSON permintaan untuk AWS Lambda fungsi.

sysmail_verify_profile_sp, sysmail_verify_account_sp, sysmail_verify_addressparams_sp

Prosedur internal yang memeriksa pengaturan.

sp_get_dbmail, sp_set_dbmail, sysmail_dbmail_xml

Prosedur internal yang tidak digunakan lagi.

Sintaks untuk prosedur yang meniru SQL Server Database Mail di Postgre SQL

`aws_sqlserver_ext.sp_send_dbmail`Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sp_send_dbmail` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_profile_name varchar = NULL::character varying,  
par_recipients text = NULL::text,  
par_copy_recipients text = NULL::text,  
par_blind_copy_recipients text = NULL::text,  
par_subject varchar = NULL::character varying,  
par_body text = NULL::text,  
par_body_format varchar = NULL::character varying,  
par_importance varchar = 'NORMAL'::character varying,
```

```
par_sensitivity varchar = 'NORMAL'::character varying,  
par_file_attachments text = NULL::text,  
par_query text = NULL::text,  
par_execute_query_database varchar = NULL::character varying,  
par_attach_query_result_as_file smallint = 0,  
par_query_attachment_filename varchar = NULL::character varying,  
par_query_result_header smallint = 1,  
par_query_result_width integer = 256,  
par_query_result_separator VARCHAR = ' '::character varying,  
par_exclude_query_output smallint = 0,  
par_append_query_error smallint = 0,  
par_query_no_truncate smallint = 0,  
par_query_result_no_padding smallint = 0,  
out par_mailitem_id integer,  
par_from_address text = NULL::text,  
par_reply_to text = NULL::text,  
out returncode integer
```

`aws_sqlserver_ext.sysmail_delete_mailitems_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_delete_mailitems_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_sent_before timestamp = NULL::timestamp without time zone,  
par_sent_status varchar = NULL::character varying,  
out returncode integer
```

`aws_sqlserver_ext.sysmail_add_profile_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_add_profile_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_profile_name varchar,  
par_description varchar = NULL::character varying,  
out par_profile_id integer,  
out returncode integer
```

`aws_sqlserver_ext.sysmail_add_account_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_add_account_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_account_name varchar
```

```
par_email_address varchar
par_display_name varchar = NULL::character varying
par_replyto_address varchar = NULL::character varying
par_description varchar = NULL::character varying
par_mailserver_name varchar = NULL::character varying
par_mailserver_type varchar = 'SMTP'::bpchar
par_port integer = 25
par_username varchar = NULL::character varying
par_password varchar = NULL::character varying
par_use_default_credentials smallint = 0
par_enable_ssl smallint = 0
out par_account_id integer
out returncode integer
```

`aws_sqlserver_ext.sysmail_add_profileaccount_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_add_profileaccount_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_profile_id integer = NULL::integer,
par_profile_name varchar = NULL::character varying,
par_account_id integer = NULL::integer,
par_account_name varchar = NULL::character varying,
par_sequence_number integer = NULL::integer,
out returncode integer
```

`aws_sqlserver_ext.sysmail_help_profile_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_help_profile_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_profile_id integer = NULL::integer,
par_profile_name varchar = NULL::character varying,
out returncode integer
```

`aws_sqlserver_ext.sysmail_update_profile_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_update_profile_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_profile_id integer = NULL::integer,
par_profile_name varchar = NULL::character varying,
```



```
par_description varchar = NULL::character varying,  
out_returncode integer
```

`aws_sqlserver_ext.sysmail_delete_profile_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_delete_profile_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_force_delete smallint = 1,  
out_returncode integer
```

`aws_sqlserver_ext.sysmail_help_account_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_help_account_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out_returncode integer
```

`aws_sqlserver_ext.sysmail_update_account_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_update_account_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
par_email_address varchar = NULL::character varying,  
par_display_name varchar = NULL::character varying,  
par_replyto_address varchar = NULL::character varying,  
par_description varchar = NULL::character varying,  
par_mailserver_name varchar = NULL::character varying,  
par_mailserver_type varchar = NULL::character varying,  
par_port integer = NULL::integer,  
par_username varchar = NULL::character varying,  
par_password varchar = NULL::character varying,  
par_use_default_credentials smallint = NULL::smallint,  
par_enable_ssl smallint = NULL::smallint,  
par_timeout integer = NULL::integer,  
par_no_credential_change smallint = NULL::smallint,  
out_returncode integer
```

`aws_sqlserver_ext.sysmail_delete_account_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_delete_account_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

`aws_sqlserver_ext.sysmail_help_profileaccount_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_help_profileaccount_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
out returncode integer
```

`aws_sqlserver_ext.sysmail_update_profileaccount_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_update_profileaccount_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,  
par_account_name varchar = NULL::character varying,  
par_sequence_number integer = NULL::integer,  
out returncode integer
```

`aws_sqlserver_ext.sysmail_delete_profileaccount_sp` Prosedur dalam paket ekstensi mengemulasi `msdb.dbo.sysmail_delete_profileaccount_sp` prosedur. Untuk informasi selengkapnya tentang prosedur Mail Database SQL Server sumber, lihat [dokumentasi teknis Microsoft](#).

```
par_profile_id integer = NULL::integer,  
par_profile_name varchar = NULL::character varying,  
par_account_id integer = NULL::integer,
```

```
par_account_name varchar = NULL::character varying,  
out returncode integer
```

Contoh untuk menggunakan prosedur yang meniru SQL Server Database Mail di Postgre SQL

Untuk mengirim email, gunakan `aws_sqlserver_ext.sp_send_dbmail` prosedur seperti yang ditunjukkan berikut.

```
PERFORM sp_send_dbmail (  
  par_profile_name := 'Administrator',  
  par_recipients := 'hello@rusgl.info',  
  par_subject := 'Automated Success Message',  
  par_body := 'The stored procedure finished'  
);
```

Contoh berikut menunjukkan cara mengirim email dengan hasil 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'  
);
```

Contoh berikut menunjukkan cara mengirim email dengan HTML kode.

```
DECLARE var_tableHTML TEXT;  
SET var_tableHTML := CONCAT(  
  '<H1>Work Order Report</H1>',  
  '<table border="1">',  
  '<tr><th>Work Order ID</th><th>Product ID</th>',  
  '<th>Name</th><th>Order Qty</th><th>Due Date</th>',  
  '<th>Expected Revenue</th></tr>',  
  '</table>'  
);  
PERFORM sp_send_dbmail (  
  par_recipients := 'hello@rusgl.info',  
  par_subject := 'Work Order List',  
  par_body := var_tableHTML,  
  par_body_format := 'HTML'  
);
```

Untuk menghapus email, gunakan `aws_sqlserver_ext.sysmail_delete_mailitems_sp` prosedur seperti yang ditunjukkan berikut.

```
DECLARE var_GETDATE datetime;
SET var_GETDATE = NOW();
PERFORM sysmail_delete_mailitems_sp (
    par_sent_before := var_GETDATE
);
```

Contoh berikut menunjukkan cara menghapus email tertua.

```
PERFORM sysmail_delete_mailitems_sp (
    par_sent_before := '31.12.2015'
);
```

Contoh berikut menunjukkan cara menghapus semua email yang tidak dapat dikirim.

```
PERFORM sysmail_delete_mailitems_sp (
    par_sent_status := 'failed'
);
```

Untuk membuat profil pengguna baru, gunakan `aws_sqlserver_ext.sysmail_add_profile_sp` prosedur seperti yang ditunjukkan berikut.

```
PERFORM sysmail_add_profile_sp (
    profile_name := 'Administrator',
    par_description := 'administrative mail'
);
```

Contoh berikut menunjukkan cara membuat profil baru dan menyimpan pengenalan profil unik dalam variabel.

```
DECLARE var_profileId INT;
SELECT par_profile_id
    FROM sysmail_add_profile_sp (
        profile_name := 'Administrator',
        par_description := ' Profile used for administrative mail.')
    INTO var_profileId;

SELECT var_profileId;
```

Untuk membuat akun email baru, gunakan `aws_sqlserver_ext.sysmail_add_account_sp` prosedur seperti yang ditunjukkan berikut.

```
PERFORM sysmail_add_account_sp (  
    par_account_name := 'Audit Account',  
    par_email_address := 'dba@rusgl.info',  
    par_display_name := 'Test Automated Mailer',  
    par_description := 'Account for administrative e-mail.',  
    par_mailserver_type := 'AWSLAMBDA'  
    par_mailserver_name := 'arn:aws:lambda:us-west-2:555555555555:function:pg_v3'  
);
```

Untuk menambahkan akun email ke profil pengguna, gunakan `aws_sqlserver_ext.sysmail_add_profileaccount_sp` prosedur seperti yang ditunjukkan berikut.

```
PERFORM sysmail_add_profileaccount_sp (  
    par_account_name := 'Administrator',  
    par_account_name := 'Audit Account',  
    par_sequence_number := 1  
);
```

Gunakan contoh kasus untuk meniru SQL Server Database Mail di Postgre SQL

Jika kode basis data sumber Anda menggunakan SQL Server Database Mail untuk mengirim email, Anda dapat menggunakan paket AWS SCT ekstensi untuk mengonversi kode ini ke SQL Postgre.

Untuk mengirim email dari database Postgre SQL Anda

1. Buat dan konfigurasi AWS Lambda fungsi Anda.
2. Terapkan paket AWS SCT ekstensi.
3. Buat profil pengguna menggunakan `sysmail_add_profile_sp` fungsi seperti yang ditunjukkan berikut.
4. Buat akun email menggunakan `sysmail_add_account_sp` fungsi seperti yang ditunjukkan berikut.
5. Tambahkan akun email ini ke profil pengguna Anda menggunakan `sysmail_add_profileaccount_sp` fungsi seperti yang ditunjukkan berikut.

```
CREATE OR REPLACE FUNCTION aws_sqlserver_ext.
```

```
proc_dbmail_settings_msdb()
RETURNS void
AS
$BODY$
BEGIN
PERFORM aws_sqlserver_ext.sysmail_add_profile_sp(
    par_profile_name := 'Administrator',
    par_description := 'administrative mail'
);
PERFORM aws_sqlserver_ext.sysmail_add_account_sp(
    par_account_name := 'Audit Account',
    par_description := 'Account for administrative e-mail.',
    par_email_address := 'dba@rusgl.info',
    par_display_name := 'Test Automated Mailer',
    par_mailserver_type := 'AWSLAMBDA'
    par_mailserver_name := 'your_ARN'
);
PERFORM aws_sqlserver_ext.sysmail_add_profileaccount_sp(
    par_profile_name := 'Administrator',
    par_account_name := 'Audit Account',
    par_sequence_number := 1
);
END;
$BODY$
LANGUAGE plpgsql;
```

6. Kirim email menggunakan `sp_send_dbmail` fungsi seperti yang ditunjukkan berikut.

```
CREATE OR REPLACE FUNCTION aws_sqlserver_ext.
proc_dbmail_send_msdb()
RETURNS void
AS
$BODY$
BEGIN
PERFORM aws_sqlserver_ext.sp_send_dbmail(
    par_profile_name := 'Administrator',
    par_recipients := 'hello@rusgl.info',
    par_body := 'The stored procedure finished',
    par_subject := 'Automated Success Message'
);
END;
$BODY$
LANGUAGE plpgsql;
```

Untuk melihat informasi tentang semua profil pengguna, gunakan `sysmail_help_profile_sp` prosedur seperti yang ditunjukkan berikut.

```
SELECT FROM aws_sqlserver_ext.sysmail_help_profile_sp();
```

Contoh berikut menampilkan informasi tentang profil pengguna tertentu.

```
select from aws_sqlserver_ext.sysmail_help_profile_sp(par_profile_id := 1);
select from aws_sqlserver_ext.sysmail_help_profile_sp(par_profile_name :=
'Administrator');
```

Untuk melihat informasi tentang semua akun email, gunakan `sysmail_help_account_sp` prosedur seperti yang ditunjukkan berikut.

```
select from aws_sqlserver_ext.sysmail_help_account_sp();
```

Contoh berikut menampilkan informasi tentang akun email tertentu.

```
select from aws_sqlserver_ext.sysmail_help_account_sp(par_account_id := 1);
select from aws_sqlserver_ext.sysmail_help_account_sp(par_account_name := 'Audit
Account');
```

Untuk melihat informasi tentang semua akun email yang terkait dengan profil pengguna, gunakan `sysmail_help_profileaccount_sp` prosedur seperti yang ditunjukkan berikut.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp();
```

Contoh berikut memfilter catatan berdasarkan pengenal, nama profil, atau nama akun.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_profile_id := 1);
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_profile_id := 1,
par_account_id := 1);
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_profile_name :=
'Administrator');
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp(par_account_name := 'Audit
Account');
```

Untuk mengubah nama atau deskripsi profil pengguna, gunakan `sysmail_update_profile_sp` prosedur seperti yang ditunjukkan berikut.

```
select aws_sqlserver_ext.sysmail_update_profile_sp(  
    par_profile_id := 2,  
    par_profile_name := 'New profile name'  
);
```

Untuk mengubah pengaturan akun email, gunakan `ysmail_update_account_sp` prosedur seperti yang ditunjukkan berikut.

```
select from aws_sqlserver_ext.sysmail_update_account_sp (  
    par_account_name := 'Audit Account',  
    par_mailserver_name := 'arn:aws:lambda:region:XXXXXXXXXXXX:function:func_test',  
    par_mailserver_type := 'AWSLAMBDA'  
);
```

Migrasi dari SQL Server ke Amazon RDS untuk SQL Server dengan AWS Schema Conversion Tool

Beberapa hal yang perlu dipertimbangkan saat memigrasikan skema dan kode SQL Server ke Amazon RDS untuk SQL Server:

- AWS SCT dapat mengonversi Agen SQL Server untuk menyediakan jadwal, peringatan, dan pekerjaan pada instans Amazon RDS untuk SQL Server DB. Setelah konversi, Anda dapat menggunakan instans Amazon RDS for SQL Server DB dengan SQL Server Reporting Services (SSRS), SQL Server Analysis Services (SSAS), dan SQL Server Integration Services (SSIS).
- Amazon RDS saat ini tidak mendukung Broker Layanan SQL Server atau SQL titik akhir T tambahan yang mengharuskan Anda menjalankan `CREATE ENDPOINT` perintah.
- Amazon RDS memiliki dukungan terbatas untuk server tertaut. Saat mengonversi kode aplikasi SQL Server yang menggunakan server tertaut, AWS SCT konversi kode aplikasi. Namun, pastikan untuk meninjau perilaku objek yang menggunakan server tertaut sebelum Anda menjalankan kode yang dikonversi.
- Selalu aktif digunakan.
- Laporan AWS SCT penilaian menyediakan metrik server untuk konversi. Metrik tentang instance SQL Server Anda ini mencakup yang berikut:
 - Pencermiran data digunakan.
 - SQL Pengiriman Log Server dikonfigurasi.
 - Failover cluster digunakan.

- Database Mail dikonfigurasi.
- Layanan Pencarian Teks Lengkap digunakan. Amazon RDS untuk SQL Server memiliki pencarian teks lengkap terbatas, dan tidak mendukung pencarian semantik.
- Layanan Kualitas Data (DQS) diinstal. Amazon RDS tidak mendukung, DQS jadi kami sarankan Anda menginstal SQL Server di EC2 instans Amazon.

Hak istimewa RDS untuk SQL Server sebagai target

Untuk bermigrasi ke RDS untuk SQL Server, buat pengguna database dan kemudian berikan hak istimewa yang diperlukan untuk setiap database. Anda dapat menggunakan contoh kode berikut.

```
CREATE LOGIN user_name WITH PASSWORD 'your_password';

USE db_name
CREATE USER user_name FOR LOGIN user_name
GRANT VIEW DEFINITION TO user_name
GRANT VIEW DATABASE STATE TO user_name
GRANT CREATE SCHEMA TO user_name;
GRANT CREATE TABLE TO user_name;
GRANT CREATE VIEW TO user_name;
GRANT CREATE TYPE TO user_name;
GRANT CREATE DEFAULT TO user_name;
GRANT CREATE FUNCTION TO user_name;
GRANT CREATE PROCEDURE TO user_name;
GRANT CREATE ASSEMBLY TO user_name;
GRANT CREATE AGGREGATE TO user_name;
GRANT CREATE FULLTEXT CATALOG TO user_name;
GRANT CREATE SYNONYM TO user_name;
GRANT CREATE XML SCHEMA COLLECTION TO user_name;
```

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

Sumber gudang data untuk AWS Schema Conversion Tool

AWS SCT dapat mengonversi skema untuk gudang data sumber berikut ke target yang didukung. Untuk informasi tentang izin, koneksi, dan apa yang AWS SCT dapat dikonversi untuk digunakan dengan database target atau gudang data, lihat detail berikut ini.

Topik

- [Menghubungkan Amazon Redshift dengan AWS Schema Conversion Tool](#)
- [Menghubungkan Azure Synapse Analytics dengan AWS Schema Conversion Tool](#)
- [Menghubungkan ke Google BigQuery dengan AWS Schema Conversion Tool](#)
- [Menghubungkan Database Greenplum dengan AWS Schema Conversion Tool](#)
- [Menghubungkan ke Netezza dengan AWS Schema Conversion Tool](#)
- [Menghubungkan Oracle Data Warehouse dengan AWS SCT](#)
- [Menghubungkan ke gudang data Snowflake dengan AWS Schema Conversion Tool](#)
- [Menghubungkan ke Gudang Data SQL Server dengan AWS Schema Conversion Tool](#)
- [Menghubungkan ke Gudang Data Teradata dengan AWS Schema Conversion Tool](#)
- [Menghubungkan database AWS Schema Conversion Tool ke Vertica](#)

Menghubungkan Amazon Redshift dengan AWS Schema Conversion Tool

Anda dapat menggunakannya AWS SCT untuk mengoptimalkan cluster Amazon Redshift Anda. AWS SCT memberi Anda rekomendasi tentang pemilihan kunci distribusi dan pengurutan untuk kluster Amazon Redshift Anda. Anda dapat mempertimbangkan proyek pengoptimalan Amazon Redshift sebagai AWS SCT proyek dengan sumber dan target yang menunjuk ke berbagai kluster Amazon Redshift.

Hak istimewa untuk Amazon Redshift sebagai database sumber

Hak istimewa berikut diperlukan untuk menggunakan Amazon Redshift sebagai sumber:

- USAGEPADA SCHEMA *<schema_name>*
- SELECTDI ALL TABLES DALAM SCHEMA *<schema_name>*
- SELECTPADA PG_ .PG_ CATALOG STATISTIC
- SELECTPADA SVV _ TABLE _ INFO
- SELECTPADA TABLE STV _ BLOCKLIST
- SELECTPADA TABLE STV _ TBL _ PERM
- SELECTPADA SYS _ SERVERLESS _ USAGE
- SELECTPADA PG_ _ DATABASE INFO
- SELECTPADA PG_ STATISTIC

Dalam contoh sebelumnya, ganti `<schema_name>` placeholder dengan nama skema sumber.

Untuk hak istimewa yang diperlukan untuk Amazon Redshift sebagai target, lihat. [Izin untuk Amazon Redshift sebagai target](#)

Menghubungkan ke Amazon Redshift sebagai sumber

Gunakan prosedur berikut untuk menyambung ke database sumber Amazon Redshift Anda dengan file. AWS Schema Conversion Tool

Untuk terhubung ke database sumber Amazon Redshift

1. Di dalam AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Amazon Redshift, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat. [Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

- Untuk memasukkan informasi koneksi untuk database sumber Amazon Redshift, gunakan petunjuk berikut:

Parameter	Tindakan
Nama server	Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.

Parameter	Tindakan
Basis Data	Masukkan nama database Amazon Redshift.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada SSLtab:</p> <ul style="list-style-type: none">• Verifikasi sertifikat server: Pilih opsi ini untuk memverifikasi sertifikat server dengan menggunakan toko kepercayaan.• Toko kepercayaan: Lokasi toko kepercayaan yang berisi sertifikat. Agar lokasi ini muncul di sini, pastikan untuk menambahkannya di pengaturan Global. <p>Untuk informasi selengkapnya tentang SSL dukungan untuk Amazon Redshift, lihat Mengonfigurasi opsi keamanan untuk koneksi.</p>
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.

Parameter	Tindakan
Jalur pengemudi Redshift	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

Pengaturan pengoptimalan Amazon Redshift

Untuk mengedit setelan optimasi Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Amazon Redshift, lalu pilih Amazon Redshift — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk optimasi Amazon Redshift.

Pengaturan pengoptimalan Amazon Redshift AWS SCT termasuk opsi untuk hal-hal berikut:

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

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

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

- Untuk mengatur jumlah maksimum tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke kluster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) dalam Panduan Manajemen Amazon Redshift.

AWS SCT mengonversi semua tabel sumber Anda, meskipun jumlah tabel lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota kluster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka akan AWS SCT menampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memilih strategi migrasi.

AWS merekomendasikan penggunaan cluster yang berbeda sebagai sumber dan target untuk proyek optimasi Anda. Sebelum memulai proses pengoptimalan Amazon Redshift, Anda membuat salinan cluster Amazon Redshift sumber Anda. Anda dapat menyertakan data sumber Anda ke dalam salinan ini atau membuat cluster kosong.

Untuk strategi Migrasi, pilih Migrasi ke salinan untuk menyertakan data dari kluster sumber Anda di kluster target.

Untuk strategi Migrasi, pilih Migrasi ke papan tulis bersih untuk meninjau saran pengoptimalan. Setelah Anda menerima saran ini, migrasi data sumber Anda ke kluster target.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift](#).

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk KEY kolom. Anda dapat memilih opsi ini hanya ketika Anda memilih opsi Gunakan pengkodean kompresi.

- Untuk bekerja dengan optimasi tabel otomatis.

Optimalisasi tabel otomatis adalah proses self-tuning di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk menggunakan hanya pada optimasi tabel otomatis, pilih strategi Optimasi di panel kiri. Kemudian pilih Gunakan penyetelan tabel otomatis Amazon Redshift, dan pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi optimasi, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Gunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilainya (0-100). Nilai-nilai ini menentukan bobot setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan mempengaruhi pilihan distribusi dan kunci pengurutan. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimasi konversi.

- Untuk Batas kolom kunci Urutkan, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kemunculan nilai yang paling umum dengan jumlah total catatan.

- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1-100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih pengguna statistik, pilih pengguna database yang statistik kuerinya ingin Anda analisis.

Menghubungkan Azure Synapse Analytics dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Azure Synapse Analytics ke Amazon Redshift.

Hak istimewa untuk Azure Synapse Analytics sebagai database sumber

Hak istimewa berikut diperlukan untuk menggunakan gudang data Azure Synapse Analytics sebagai sumber:

- VIEW DEFINITION
- VIEW DATABASE STATE

Terapkan hak istimewa untuk setiap database yang skema Anda konversi.

Menghubungkan ke Azure Synapse Analytics sebagai sumber

Gunakan prosedur berikut untuk menyambung ke gudang data Azure Synapse Analytics Anda dengan AWS Schema Conversion Tool

Untuk terhubung ke gudang data Azure Synapse Analytics sebagai sumber

1. Di dalam AWS Schema Conversion Tool, pilih Tambahkan sumber.
2. Pilih Azure Synapse Analytics, lalu pilih Berikutnya.

Kotak dialog Tambah sumber muncul.

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

1. Untuk AWS Rahasia, pilih nama rahasianya.
2. Pilih Isi untuk secara otomatis mengisi semua nilai di kotak dialog koneksi database dari Secrets Manager.

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

- Untuk memasukkan informasi koneksi untuk gudang data Azure Synapse Analytics secara manual, gunakan petunjuk berikut:

Parameter	Tindakan
Nama server	Masukkan nama Layanan Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
SQLkolam	Masukkan nama SQL kolam Azure.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada SSLtab:</p> <ul style="list-style-type: none"> • Sertifikat server kepercayaan: Pilih opsi ini untuk mempercayai sertifikat server. • Toko kepercayaan: Toko kepercayaan yang Anda atur di pengaturan Global.

Parameter	Tindakan
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa memasukkan kata sandi.

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

Pengaturan konversi Azure Synapse Analytics ke Amazon Redshift

Untuk mengedit pengaturan konversi Azure Synapse Analytics ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Azure Synapse, lalu pilih Azure Synapse — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Azure Synapse Analytics ke Amazon Redshift.

Pengaturan konversi Azure Synapse Analytics ke Amazon Redshift AWS SCT termasuk opsi untuk hal-hal berikut:

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

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

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

- Untuk mengatur jumlah maksimum tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke kluster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) dalam Panduan Manajemen Amazon Redshift.

AWS SCT mengonversi semua tabel sumber Anda, meskipun ini lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota kluster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka akan AWS SCT menampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasikan partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan UNION ALL tampilan dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk tabel sumber tunggal.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasikan setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang menyertakan data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke cluster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift](#).

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi

ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk KEY kolom. Anda dapat memilih opsi ini hanya ketika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Azure Synapse Analytics ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Azure Synapse Analytics ke Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Azure Synapse, lalu pilih Azure Synapse — Amazon Redshift. Di panel kiri, pilih Strategi optimasi. AWS SCT menampilkan setelan pengoptimalan konversi untuk konversi Azure Synapse Analytics ke Amazon Redshift.

Pengaturan pengoptimalan konversi Azure Synapse Analytics ke Amazon Redshift AWS SCT termasuk opsi untuk hal-hal berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses self-tuning di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi optimasi, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Gunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilainya (0-100). Nilai-nilai ini menentukan bobot setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan mempengaruhi pilihan distribusi dan kunci pengurutan. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimasi konversi.

- Untuk Batas kolom kunci Urutkan, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kemunculan nilai yang paling umum dengan jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1-100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih pengguna statistik, pilih pengguna database yang ingin Anda analisis statistik kueri.

Juga, pada tab Strategi optimasi, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menghubungkan ke Google BigQuery dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Amazon BigQuery Redshift.

Hak istimewa untuk BigQuery sebagai sumber

Untuk menggunakan gudang BigQuery data sebagai sumber AWS SCT, buat akun layanan. Di Google Cloud, aplikasi menggunakan akun layanan untuk melakukan API panggilan resmi. Akun layanan berbeda dari akun pengguna. Untuk informasi selengkapnya, lihat [Akun layanan](#) di dokumentasi Google Cloud Identity and Access Management.

Pastikan Anda memberikan peran berikut ke akun layanan Anda:

- BigQuery Admin
- Storage Admin

BigQuery Admin Peran tersebut memberikan izin untuk mengelola semua sumber daya dalam proyek. AWS SCT menggunakan peran ini untuk memuat BigQuery metadata Anda dalam proyek migrasi.

Storage Admin Peran tersebut memberikan kontrol penuh atas objek data dan ember. Anda dapat menemukan peran ini di bawah Cloud Storage. AWS SCT menggunakan peran ini untuk mengekstrak data Anda BigQuery dan kemudian memuatnya ke Amazon Redshift.

Untuk membuat file kunci akun layanan

1. Masuk ke konsol manajemen Google Cloud di <https://console.cloud.google.com/>.
2. Pada [BigQuery API](#) halaman, pilih Aktifkan. Lewati langkah ini jika Anda melihat API Diaktifkan.
3. Pada halaman [Akun layanan](#), pilih proyek Anda, lalu pilih Buat akun layanan.
4. Pada halaman detail akun Layanan, masukkan nilai deskriptif untuk nama akun Layanan. Pilih Buat dan lanjutkan. Berikan akses akun layanan ini ke halaman proyek terbuka.
5. Untuk Pilih peran, pilih BigQuery, lalu pilih BigQuery Admin.
6. Pilih Tambahkan peran lain. Untuk Memilih peran, pilih Cloud Storage, lalu pilih Admin Penyimpanan.
7. Pilih Lanjutkan, lalu pilih Selesai.
8. Pada halaman [Akun layanan](#), pilih akun layanan yang Anda buat.
9. Pilih Keys, lalu pilih Create new key for Add key.
10. Pilih JSON, lalu pilih Buat. Pilih folder untuk menyimpan kunci pribadi Anda atau pilih folder default untuk diunduh di browser Anda.

Untuk mengekstrak data dari gudang BigQuery data, AWS SCT gunakan folder bucket Google Cloud Storage. Buat bucket ini sebelum Anda memulai migrasi data. Masukkan path ke folder bucket Google Cloud Storage Anda di kotak dialog Create Local task. Untuk informasi selengkapnya, lihat [Membuat, menjalankan, dan memantau AWS SCT tugas](#).

Menghubungkan ke BigQuery sebagai sumber

Gunakan prosedur berikut untuk terhubung ke BigQuery proyek sumber Anda dengan AWS Schema Conversion Tool.

Untuk terhubung ke gudang data BigQuery sumber

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

Kotak dialog Tambah sumber muncul.

3. Untuk nama Koneksi, masukkan nama untuk BigQuery proyek Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
4. Untuk jalur Kunci, masukkan jalur ke file kunci akun layanan. Untuk informasi selengkapnya tentang membuat file ini, lihat [Hak istimewa untuk BigQuery sebagai sumber](#).
5. Pilih Uji Koneksi untuk memverifikasi bahwa AWS SCT dapat terhubung ke BigQuery proyek sumber Anda.
6. Pilih Connect untuk terhubung ke BigQuery proyek sumber Anda.

Keterbatasan penggunaan BigQuery sebagai sumber untuk AWS SCT

Batasan berikut berlaku saat menggunakan BigQuery sebagai sumber untuk AWS SCT:

- AWS SCT tidak mendukung konversi subkueri dalam fungsi analitik.
- Anda tidak dapat menggunakan AWS SCT untuk mengonversi BigQuery `SELECT AS STRUCT` dan `SELECT AS VALUE` pernyataan.
- AWS SCT tidak mendukung konversi jenis fungsi berikut:
 - Perkiraan agregat
 - Bit
 - Debugging
 - Kueri federasi
 - Geografi
 - Hash
 - Matematika
 - Bersih

- Agregat statistik
- UUID
- AWS SCT menyediakan dukungan terbatas untuk konversi fungsi string.
- AWS SCT tidak mendukung konversi UNNEST operator.
- Anda tidak dapat mengonversi operasi gabungan yang berkorelasi di AWS SCT.
- AWS SCT tidak mendukung konversi QUALIFY,, WINDOWLIMIT, dan OFFSET klausa.
- Anda tidak dapat menggunakan AWS SCT untuk mengonversi ekspresi tabel umum rekursif.
- AWS SCT tidak mendukung konversi INSERT pernyataan dengan subkueri di dalam VALUES klausa.
- AWS SCT tidak mendukung konversi UPDATE pernyataan untuk bidang bersarang dan catatan berulang.
- Anda tidak dapat menggunakan AWS SCT untuk mengkonversi STRUCT dan tipe ARRAY data.

BigQuery ke pengaturan konversi Amazon Redshift

Untuk mengedit BigQuery ke pengaturan konversi Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Google BigQuery, lalu pilih Google BigQuery — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia BigQuery untuk konversi Amazon Redshift.

BigQuery ke setelan konversi Amazon Redshift di AWS SCT menyertakan opsi untuk hal berikut:

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

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

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

- Untuk mengatur jumlah maksimum tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) dalam Panduan Manajemen Amazon Redshift.

AWS SCT mengonversi semua tabel sumber Anda, meskipun ini lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka akan AWS SCT menampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift](#).

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk KEY kolom. Anda dapat memilih opsi ini hanya ketika Anda memilih opsi Gunakan pengkodean kompresi.

BigQuery ke pengaturan pengoptimalan konversi Amazon Redshift

Untuk mengedit BigQuery ke pengaturan pengoptimalan konversi Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Google BigQuery, lalu pilih Google BigQuery — Amazon Redshift. Di panel kiri, pilih Strategi optimasi. AWS SCT menampilkan pengaturan pengoptimalan konversi BigQuery untuk konversi Amazon Redshift.

BigQuery ke setelan pengoptimalan konversi Amazon Redshift di AWS SCT menyertakan opsi untuk hal berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses self-tuning di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi optimasi, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Gunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilainya (0-100). Nilai-nilai ini menentukan bobot setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan mempengaruhi pilihan distribusi dan kunci pengurutan. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimasi konversi.

- Untuk Batas kolom kunci Urutkan, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kemunculan nilai yang paling umum dengan jumlah total catatan.

- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1-100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih pengguna statistik, pilih pengguna database yang ingin Anda analisis statistik kueri.

Juga, pada tab Strategi optimasi, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menghubungkan Database Greenplum dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Greenplum Database ke Amazon Redshift.

Hak istimewa untuk Database Greenplum sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan Database Greenplum sebagai sumber:

- CONNECTPADA DATABASE *<database_name>*
- USAGEPADA SCHEMA *<schema_name>*
- SELECTPADA *<schema_name>.<table_name>*
- SELECTPADA SEQUENCE *<schema_name>.<sequence_name>*

Pada contoh sebelumnya, ganti placeholder sebagai berikut:

- Ganti *database_name* dengan nama database sumber.
- Ganti *schema_name* dengan nama skema sumber.
- Ganti *table_name* dengan nama tabel sumber.
- Ganti *sequence_name* dengan nama nama urutan.

Menghubungkan ke Greenplum Database sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Greenplum Anda dengan AWS SCT.

Untuk terhubung ke database sumber Greenplum

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

- Untuk memasukkan kredensi database sumber Greenplum secara manual, gunakan petunjuk berikut:

Parameter	Tindakan
Nama server	Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis Data	Masukkan nama database Greenplum.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara</p>

Parameter	Tindakan
	default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada SSLtab:</p> <ul style="list-style-type: none"> • Verifikasi sertifikat server: Pilih opsi ini untuk memverifikasi sertifikat server dengan menggunakan toko kepercayaan. • Toko kepercayaan: Lokasi toko kepercayaan yang berisi sertifikat.
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.
Jalur driver Database Greenplum	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

Pengaturan konversi Greenplum ke Amazon Redshift

Untuk mengedit pengaturan konversi Greenplum ke Amazon Redshift, pilih Pengaturan AWS SCT di, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Greenplum, lalu pilih Greenplum — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Greenplum ke Amazon Redshift.

Pengaturan konversi Greenplum ke Amazon Redshift AWS SCT di menyertakan opsi untuk hal berikut:

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

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

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

- Untuk mengatur jumlah maksimum tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) dalam Panduan Manajemen Amazon Redshift.

AWS SCT mengonversi semua tabel sumber Anda, meskipun ini lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka akan AWS SCT menampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasikan partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan UNION ALL tampilan dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk tabel sumber tunggal.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasikan setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang menyertakan data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke cluster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift](#).

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk KEY kolom. Anda dapat memilih opsi ini hanya ketika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Greenplum ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Greenplum ke Amazon Redshift, pilih Pengaturan AWS SCT di, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Greenplum, lalu pilih Greenplum — Amazon Redshift. Di panel kiri, pilih Strategi optimasi. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Greenplum ke Amazon Redshift.

Pengaturan pengoptimalan konversi Greenplum ke Amazon Redshift AWS SCT di menyertakan opsi untuk hal berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses self-tuning di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi optimasi, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Gunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilainya (0-100). Nilai-nilai ini menentukan bobot setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan mempengaruhi pilihan distribusi dan kunci pengurutan. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimasi konversi.

- Untuk Batas kolom kunci Urutkan, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kemunculan nilai yang paling umum dengan jumlah total catatan.

- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1-100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih pengguna statistik, pilih pengguna database yang ingin Anda analisis statistik kueri.

Juga, pada tab Strategi optimasi, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menghubungkan ke Netezza dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Netezza ke Amazon Redshift.

Hak istimewa untuk Netezza sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan Netezza sebagai sumber:

- pilih pada tampilan `system.definition_schema.system`
- pilih pada tabel `system.definition_schema.system`
- pilih pada tabel `system.definition_schema.management`
- daftar di `<database_name>`
- daftar di `<schema_name>`
- daftar di `<database_name>.all.table`
- daftar di `<database_name>.all.external tabel.`
- daftar di `<database_name>.semua.tampilan`
- daftar di `<database_name>.all.materialized tampilan`
- daftar di `<database_name>.all.procedure`
- daftar di `<database_name>.all.sequence`
- daftar di `<database_name>.all.fungsi`
- daftar di `<database_name>.semua.agregat`

Pada contoh sebelumnya, ganti placeholder sebagai berikut:

- Ganti *database_name* dengan nama database sumber.
- Ganti *schema_name* dengan nama skema sumber.

AWS SCT membutuhkan akses ke tabel dan tampilan sistem berikut. Anda dapat memberikan akses ke objek-objek ini alih-alih memberikan akses ke `system.definition_schema.system view` dan `system.definition_schema.system tables` dalam daftar sebelumnya.

- pilih pada `system.definition_schema. _t_agregat`
- pilih pada `system.definition_schema. _t_class`
- pilih pada `system.definition_schema. _t_kendala`
- pilih pada `system.definition_schema. _t_const_relattr`
- pilih pada `system.definition_schema. _t_database`
- pilih pada `system.definition_schema. _t_grpobj_priv`
- pilih pada `system.definition_schema. _t_grpusr`
- pilih pada `system.definition_schema. _t_hist_config`
- pilih pada `system.definition_schema. _t_objek`
- pilih pada `system.definition_schema. _t_object_classes`
- pilih pada `system.definition_schema. _t_proc`
- pilih pada `system.definition_schema. _t_type`
- pilih pada `system.definition_schema. _t_pengguna`
- pilih pada `system.definition_schema. _t_usrobj_priv`
- pilih pada `system.definition_schema. _vt_urutan`
- pilih pada `system.definition_schema. _v_agregat`
- pilih pada `system.definition_schema. _v_constraint_tergantung`
- pilih pada `system.definition_schema. _v_basis data`
- pilih pada `system.definition_schema. _v_tipe data`
- pilih pada `system.definition_schema. _v_dslice`
- pilih pada `system.definition_schema. _v_fungsi`
- pilih pada `system.definition_schema. _v_grup`
- pilih pada `system.definition_schema. _v_obj_relasi`

- pilih pada `system.definition_schema. _v_obj_relasi_xdb`
- pilih pada `system.definition_schema. _v_prosedur`
- pilih pada `system.definition_schema. _v_hubungan_kolom`
- pilih pada `system.definition_schema. _hubungan_keydata`
- pilih pada `system.definition_schema. _v_relobjclasses`
- pilih pada `system.definition_schema. _v_schema_xdb`
- pilih pada `system.definition_schema. _v_urutan`
- pilih pada `system.definition_schema. _v_sinonim`
- pilih pada `system.definition_schema. _v_system_info`
- pilih pada `system.definition_schema. _v_sys_kendala`
- pilih pada `system.definition_schema. _v_sys_object_dslice_info`
- pilih pada `system.definition_schema. _v_sys_pengguna`
- pilih pada `system.definition_schema. _v_tabel`
- pilih pada `system.definition_schema. _v_table_kendala`
- pilih pada `system.definition_schema. _v_table_dist_map`
- pilih pada `system.definition_schema. _v_table_organize_column`
- pilih pada `system.definition_schema. _v_table_storage_stat`
- pilih pada `system.definition_schema. _v_pengguna`
- pilih pada `system.definition_schema. _v_lihat`
- pilih pada `system.information_schema. _v_hubungan_kolom`
- pilih pada `system.information_schema. _v_tabel`
- pilih pada `$hist_column_access_*`

Menghubungkan ke Netezza sebagai sumber

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

Untuk terhubung ke database sumber Netezza

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

Parameter	Tindakan
Nama server	Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Dengan mengaktif

Parameter	Tindakan
	<p>kan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.</p>
<p>Jalur pengemudi Netezza</p>	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

Mengonfigurasi replikasi data yang sedang berlangsung

Setelah mengonversi skema database Netezza dan menerapkannya ke database Amazon Redshift, Anda dapat memigrasi data dengan agen ekstraksi data. AWS SCT Agen mengekstrak data Anda dan mengunggahnya ke bucket Amazon S3 Anda. Anda kemudian dapat menggunakannya AWS SCT untuk menyalin data dari Amazon S3 ke Amazon Redshift.

Jika data dalam database sumber Anda berubah selama proses migrasi, Anda dapat menangkap perubahan yang sedang berlangsung dengan agen ekstraksi AWS SCT data Anda. Kemudian Anda dapat mereplikasi perubahan yang sedang berlangsung ini di database target Anda setelah Anda menyelesaikan migrasi data awal. Proses ini disebut replikasi data berkelanjutan atau change data capture (CDC).

Untuk mengonfigurasi replikasi data yang sedang berlangsung untuk migrasi dari Netezza ke Amazon Redshift

1. Di database sumber Anda, buat database riwayat. Anda dapat menggunakan contoh kode berikut di antarmuka baris perintah Netezza ()CLI.

```
nzhistcreatedb -d history_database_name -t query -v 1 -u load_user -o histdb_owner
-p your_password
```

Pada contoh sebelumnya, ganti *history_database_name* dengan nama database riwayat Anda. Selanjutnya, ganti *load_user* dengan nama pengguna yang telah Anda tentukan untuk memuat data riwayat ke database. Kemudian, ganti *histdb_owner* dengan nama pengguna yang telah Anda definisikan sebagai pemilik database riwayat. Pastikan Anda telah membuat pengguna ini dan memberikan CREATE DATABASE izin. Akhirnya, ganti *your_password* dengan kata sandi yang aman.

2. Konfigurasi pencatatan riwayat. Untuk melakukannya, gunakan contoh kode berikut.

```
CREATE HISTORY CONFIGURATION history_configuration_name HISTTYPE QUERY
DATABASE history_database_name USER load_user PASSWORD your_password COLLECT
PLAN, COLUMN
LOADINTERVAL 1 LOADMINTHRESHOLD 0 LOADMAXTHRESHOLD 0 STORAGELIMIT 25
LOADRETRY 2 VERSION 1;
```

Pada contoh sebelumnya, ganti *history_configuration_name* and *history_database_name* dengan nama-nama konfigurasi riwayat Anda dan database riwayat Anda. Selanjutnya, ganti *load_user* dengan nama pengguna yang telah Anda tentukan untuk memuat data riwayat ke database. Kemudian, ganti *your_password* dengan kata sandi yang aman.

3. Berikan izin baca untuk semua tabel dalam database riwayat. Anda dapat menggunakan contoh kode berikut untuk memberikan SELECT izin.

```
GRANT SELECT ON history_database_name.ALL.TABLE TO your_user;
```

Pada contoh sebelumnya, ganti *history_database_name* dengan nama database riwayat Anda. Selanjutnya, ganti *your_user* dengan nama pengguna dengan izin minimal untuk bekerja dengan database Netezza Anda. Anda menggunakan kredensi pengguna database ini di. AWS SCT

4. Kumpulkan statistik untuk setiap tabel dalam skema sumber Anda untuk mendapatkan informasi tentang kardinalitas kolom. Anda dapat menggunakan perintah berikut untuk menghasilkan statistik dalam database riwayat Anda.

```
GENERATE STATISTICS on "schema_name".table_name";
```

Pada contoh sebelumnya, ganti *schema_name* and *table_name* dengan nama skema dan tabel database Anda.

5. Pastikan Anda menyelesaikan prasyarat dengan menjalankan kueri berikut:

```
SELECT COUNT(*)  
FROM history_database_name.history_schema_name."$hist_column_access_N";
```

Pada contoh sebelumnya, ganti *history_database_name* and *history_schema_name* dengan nama database dan skema riwayat Anda. Selanjutnya, ganti *N* dengan nomor versi database riwayat Anda. Untuk informasi selengkapnya tentang versi database riwayat, lihat Dokumentasi [IBMNetezza](#).

6. Instal agen ekstraksi data Anda. Untuk informasi selengkapnya, lihat [Instalasi agen ekstraksi](#).

Pastikan bahwa {`working.folder`} parameter dalam `settings.properties` file untuk semua instance extractor menunjuk ke folder yang sama. Dalam hal ini, ekstraktor Anda dapat mengoordinasikan CDC sesi dan menggunakan satu titik transaksi untuk semua subtugas.

7. Daftarkan agen ekstraksi data Anda. Untuk informasi selengkapnya, lihat [Mendaftarkan agen ekstraksi dengan AWS Schema Conversion Tool](#).

8. Buat CDC tugas Anda. Untuk informasi selengkapnya, lihat [Membuat, menjalankan, dan memantau AWS SCT tugas](#).

- a. Buka proyek Anda di AWS SCT. Di panel kiri, pilih tabel sumber Anda. Buka menu konteks (klik kanan), dan pilih Buat tugas lokal.
- b. Untuk nama Tugas, masukkan nama deskriptif untuk tugas migrasi data Anda.
- c. Untuk mode Migrasi, pilih Ekstrak, unggah, dan salin.
- d. Pilih AktifkanCDC.
- e. Pilih tab CDCpengaturan dan tentukan ruang lingkup dan jadwal CDC sesi.
- f. Pilih Tugas uji untuk memverifikasi bahwa Anda dapat terhubung ke folder kerja, bucket Amazon S3, dan gudang data Amazon Redshift.
- g. Pilih Buat untuk membuat tugas Anda.
- h. Pilih tab Tugas, pilih tugas Anda dari daftar, dan pilih Mulai.

9. AWS SCT Tugas menjaga konsistensi transaksional pada database target. Agen ekstraksi data mereplikasi transaksi dari sumber dalam urutan ID transaksi.

Jika Anda menghentikan salah satu sesi migrasi atau jika gagal, maka CDC pemrosesan juga berhenti.

Pengaturan konversi Netezza ke Amazon Redshift

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

Pengaturan konversi Netezza ke Amazon Redshift AWS SCT di menyertakan opsi untuk hal berikut:

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

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

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

- Untuk mengatur jumlah maksimum tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) dalam Panduan Manajemen Amazon Redshift.

AWS SCT mengonversi semua tabel sumber Anda, meskipun ini lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek

Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka akan AWS SCT menampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift](#).

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk KEY kolom. Anda dapat memilih opsi ini hanya ketika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Netezza ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Netezza ke Amazon Redshift, pilih Pengaturan AWS SCT di, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Netezza, lalu pilih Netezza - Amazon Redshift. Di panel kiri, pilih Strategi optimasi. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Netezza ke Amazon Redshift.

Pengaturan pengoptimalan konversi Netezza ke Amazon Redshift AWS SCT di menyertakan opsi untuk hal berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses self-tuning di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi optimasi, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Gunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilainya (0-100). Nilai-nilai ini menentukan bobot setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan mempengaruhi pilihan distribusi dan kunci pengurutan. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimasi konversi.

- Untuk Batas kolom kunci Urutkan, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kemunculan nilai yang paling umum dengan jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1-100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih pengguna statistik, pilih pengguna database yang ingin Anda analisis statistik kueri.

Juga, pada tab Strategi optimasi, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menghubungkan Oracle Data Warehouse dengan AWS SCT

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Oracle Data Warehouse ke Amazon Redshift atau Amazon Redshift dan digunakan dalam kombinasi. AWS Glue

Hak istimewa untuk Oracle Data Warehouse sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan Oracle Data Warehouse sebagai sumber:

- `hubungkan`
- `select_catalog_role`
- pilih kamus apa saja

Menghubungkan ke Oracle Data Warehouse sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber gudang data Oracle Anda dengan file. AWS Schema Conversion Tool

Untuk terhubung ke database sumber Oracle Data Warehouse

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

Parameter	Tindakan
Jenis	<p>Pilih jenis koneksi ke database Anda. Tergantung pada jenis Anda, berikan informasi tambahan berikut:</p> <ul style="list-style-type: none">• SID<ul style="list-style-type: none">• Nama server: Nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.• Port server: Port yang digunakan untuk terhubung ke server database sumber Anda.• Oracle SID: ID Sistem Oracle (SID). Untuk menemukan OracleSID, kirimkan kueri berikut ke database Oracle Anda: <pre>SELECT sys_context('userenv', 'instance_name') AS SID FROM dual;</pre>• Nama Layanan<ul style="list-style-type: none">• Nama server: DNS Nama atau alamat IP server basis data sumber Anda.• Port server: Port yang digunakan untuk terhubung ke server database sumber Anda.• Nama Layanan: Nama layanan Oracle untuk terhubung• TNSalias<ul style="list-style-type: none">• TNSpath file: Path ke file yang berisi informasi koneksi nama Transparent Network Substrate (TNS).• TNSfile path: TNS Alias dari file ini untuk digunakan untuk terhubung ke database sumber.• TNSsambungkan pengenal<ul style="list-style-type: none">• TNSconnect identifier: Pengidentifikasi untuk informasi TNS koneksi terdaftar.

Parameter	Tindakan
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada SSLtab:</p> <ul style="list-style-type: none">• SSLotentikasi: Pilih opsi ini untuk menggunakan SSL otentikasi untuk koneksi.• Toko kepercayaan: Lokasi toko kepercayaan yang berisi sertifikat.• Key store: Lokasi toko kunci yang berisi kunci pribadi dan sertifikat. Nilai ini diperlukan jika SSLotentikasi dipilih dan sebaliknya opsional.
Menyimpan kata sandi	<p>AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.</p>

Parameter	Tindakan
Jalur pengemudi Oracle	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

Pengaturan konversi Oracle Data Warehouse ke Amazon Redshift

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

Pengaturan AWS SCT konversi Oracle Data Warehouse ke Amazon Redshift termasuk opsi untuk hal-hal berikut:

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

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

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

- Untuk mengatur jumlah maksimum tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) dalam Panduan Manajemen Amazon Redshift.

AWS SCT mengonversi semua tabel sumber Anda, meskipun ini lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka akan AWS SCT menampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasikan partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan UNION ALL tampilan dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk tabel sumber tunggal.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasikan setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang menyertakan data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke cluster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk mengonversi fungsi pemformatan tipe data seperti TO_CHAR, TO_DATE, dan TO_NUMBER dengan elemen format datetime yang tidak didukung Amazon Redshift. Secara default, AWS SCT menggunakan fungsi paket ekstensi untuk meniru penggunaan elemen format yang tidak didukung ini dalam kode yang dikonversi.

Model format datetime di Oracle mencakup lebih banyak elemen dibandingkan dengan string format datetime di Amazon Redshift. Jika kode sumber Anda hanya menyertakan elemen format

datetime yang didukung Amazon Redshift, Anda tidak memerlukan fungsi paket ekstensi dalam kode yang dikonversi. Untuk menghindari penggunaan fungsi paket ekstensi dalam kode yang dikonversi, pilih Elemen format tipe data yang Anda gunakan dalam kode Oracle mirip dengan string format datetime di Amazon Redshift. Dalam hal ini, kode yang dikonversi bekerja lebih cepat.

Model format numerik di Oracle mencakup lebih banyak elemen dibandingkan dengan string format numerik di Amazon Redshift. Jika kode sumber Anda hanya menyertakan elemen format numerik yang didukung Amazon Redshift, Anda tidak memerlukan fungsi paket ekstensi dalam kode yang dikonversi. Untuk menghindari penggunaan fungsi paket ekstensi dalam kode yang dikonversi, pilih Elemen format numerik yang Anda gunakan dalam kode Oracle mirip dengan string format numerik di Amazon Redshift. Dalam hal ini, kode yang dikonversi bekerja lebih cepat.

- Untuk mengkonversi fungsi Oracle LEAD dan LAG analitik. Secara default, AWS SCT memunculkan item tindakan untuk masing-masing LEAD dan LAG fungsi.

Ketika kode sumber Anda tidak menggunakan nilai default untuk offset dalam fungsi ini, AWS SCT dapat meniru penggunaan fungsi ini dengan fungsi. NVL Untuk melakukannya, pilih Gunakan NVL fungsi untuk meniru perilaku Oracle LEAD dan LAG fungsi.

- Untuk meniru perilaku kunci primer dan unik di kluster Amazon Redshift Anda, pilih Emulasikan perilaku kunci primer dan unik.

Amazon Redshift tidak menerapkan kunci unik dan utama dan menggunakannya hanya untuk tujuan informasi. Jika Anda menggunakan batasan ini dalam kode Anda, maka pastikan itu AWS SCT mengemulasi perilaku mereka dalam kode yang dikonversi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift](#).

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk KEY kolom. Anda dapat memilih opsi ini hanya ketika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Oracle Data Warehouse ke Amazon Redshift

Untuk mengedit Oracle Data Warehouse ke pengaturan pengoptimalan konversi Amazon Redshift, pilih Pengaturan AWS SCT di, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Oracle, lalu pilih Oracle - Amazon Redshift. Di panel kiri, pilih Strategi optimasi. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Oracle Data Warehouse ke Amazon Redshift.

Pengaturan AWS SCT pengoptimalan konversi Oracle Data Warehouse ke Amazon Redshift termasuk opsi untuk hal-hal berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses self-tuning di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi optimasi, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Gunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilainya (0-100). Nilai-nilai ini menentukan bobot setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan mempengaruhi pilihan distribusi dan kunci pengurutan. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimasi konversi.

- Untuk Batas kolom kunci Urutkan, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kemunculan nilai yang paling umum dengan jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1-100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih pengguna statistik, pilih pengguna database yang ingin Anda analisis statistik kueri.

Juga, pada tab Strategi optimasi, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menghubungkan ke gudang data Snowflake dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Snowflake ke Amazon Redshift.

Hak istimewa untuk Snowflake sebagai database sumber

Anda dapat membuat peran dengan hak istimewa dan memberikan peran ini nama pengguna dengan menggunakan SECURITYADMIN peran dan konteks SECURITYADMIN sesi.

Contoh berikut menciptakan hak istimewa minimal dan memberikannya kepada pengguna. `min_privs`

```
create role role_name;  
grant role role_name to role sysadmin;  
grant usage on database db_name to role role_name;  
grant usage on schema db_name.schema_name to role role_name;  
grant usage on warehouse datawarehouse_name to role role_name;  
grant monitor on database db_name to role role_name;
```

```
grant monitor on warehouse datawarehouse_name to role role_name;  
grant select on all tables in schema db_name.schema_name to role role_name;  
grant select on future tables in schema db_name.schema_name to role role_name;  
grant select on all views in schema db_name.schema_name to role role_name;  
grant select on future views in schema db_name.schema_name to role role_name;  
grant select on all external tables in schema db_name.schema_name to role role_name;  
grant select on future external tables in schema db_name.schema_name to role role_name;  
grant usage on all sequences in schema db_name.schema_name to role role_name;  
grant usage on future sequences in schema db_name.schema_name to role role_name;  
grant usage on all functions in schema db_name.schema_name to role role_name;  
grant usage on future functions in schema db_name.schema_name to role role_name;  
grant usage on all procedures in schema db_name.schema_name to role role_name;  
grant usage on future procedures in schema db_name.schema_name to role role_name;  
create user min_privs password='real_user_password'  
DEFAULT_ROLE = role_name DEFAULT_WAREHOUSE = 'datawarehouse_name';  
grant role role_name to user min_privs;
```

Pada contoh sebelumnya, ganti placeholder sebagai berikut:

- Ganti *role_name* dengan nama peran dengan hak istimewa hanya-baca.
- Ganti *db_name* dengan nama database sumber.
- Ganti *schema_name* dengan nama skema sumber.
- Ganti *nama datawarehouse_name* dengan nama gudang data yang diperlukan.
- Ganti *min_privs* dengan nama pengguna yang memiliki hak istimewa minimal.

DEFAULT_WAREHOUSEParameter DEFAULT_ROLE dan sensitif terhadap kunci.

Mengkonfigurasi akses aman ke Amazon S3

Kebijakan keamanan dan manajemen akses untuk bucket Amazon S3 memungkinkan Snowflake mengakses, membaca data, dan menulis data ke bucket S3. Anda dapat mengonfigurasi akses aman ke bucket Amazon S3 pribadi menggunakan jenis objek SnowflakeSTORAGE INTEGRATION. Objek integrasi penyimpanan Snowflake mendelegasikan tanggung jawab otentikasi ke identitas Snowflake dan entitas manajemen akses.

Untuk informasi selengkapnya, lihat [Mengonfigurasi Integrasi Penyimpanan Snowflake untuk Mengakses Amazon S3 dalam dokumentasi Snowflake](#).

Menghubungkan ke Snowflake sebagai sumber

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

Untuk terhubung ke database sumber Snowflake

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

- Untuk memasukkan informasi koneksi gudang data sumber Snowflake secara manual, gunakan instruksi berikut:

Parameter	Tindakan
Nama server	Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis Data	Masukkan nama database Snowflake.
Nama pengguna dan Kata Sandi	Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.

Parameter	Tindakan
	AWS SCT menyimpan kata sandi Anda dalam format terenkripsi hanya jika Anda memintanya secara eksplisit.
Gunakan SSL	<p>Pilih opsi ini jika Anda ingin menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada SSLtab:</p> <ul style="list-style-type: none"> • Jalur kunci pribadi: Lokasi kunci pribadi. • Passphrase: Passphrase untuk kunci privat. <p>Untuk informasi selengkapnya tentang SSL dukungan untuk Snowflake, lihat Mengkonfigurasi opsi keamanan untuk koneksi.</p>
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Jika Anda mengatur opsi ini, Anda dapat menyimpan kata sandi database. Melakukan ini berarti Anda dapat terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.
Jalur pengemudi kepingan salju	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

Keterbatasan untuk Snowflake sebagai sumber

Berikut ini adalah batasan saat menggunakan Snowflake sebagai sumber untuk: AWS SCT

- Pengidentifikasi objek harus unik dalam konteks jenis objek dan objek induk:

Basis Data

Pengidentifikasi skema harus unik dalam database.

Skema

Pengidentifikasi objek seperti untuk tabel dan tampilan harus unik dalam skema.

Tabel/Tampilan

Pengidentifikasi kolom harus unik di dalam tabel.

- Jumlah maksimum tabel untuk tipe node cluster besar dan xlarge adalah 9.900. Untuk tipe node cluster 8xlarge, jumlah maksimum tabel adalah 100.000. Batas tersebut mencakup tabel sementara, baik yang ditentukan pengguna maupun yang dibuat oleh Amazon Redshift selama pemrosesan kueri atau pemeliharaan sistem. Untuk informasi selengkapnya, lihat [kuota Amazon Redshift di Panduan Manajemen Cluster Amazon Redshift](#).
- Untuk prosedur tersimpan, jumlah maksimum argumen input dan output adalah 32.

Tipe data sumber untuk Snowflake

Berikut ini, Anda dapat menemukan tipe data sumber Snowflake yang didukung saat menggunakan AWS SCT dan pemetaan default ke target Amazon Redshift.

Jenis data kepingan salju	Tipe data Amazon Redshift
NUMBER	NUMERIC(38)
NUMBER(p)	Jika p adalah =< 4, maka SMALLINT Jika p adalah => 5 dan =< 9, maka INTEGER Jika p adalah => 10 dan =< 18, maka BIGINT Jika p adalah => 19 maka NUMERIC (p)
NUMBER(p, 0)	Jika p adalah =< 4, maka SMALLINT

Jenis data kepingan salju	Tipe data Amazon Redshift
	<p>Jika p adalah => 5 dan =< 9, maka INTEGER</p> <p>Jika p adalah => 10 dan =< 18, maka BIGINT</p> <p>Jika p adalah => 19 maka: NUMERIC (p, 0)</p>
NUMBER(p, s)	<p>Jika p adalah => 1 dan =< 38, dan jika s adalah => 1 dan =< 37, maka</p> <p>NUMERIC(p, s)</p>
FLOAT	FLOAT
<p>TEXT</p> <p>Karakter unicode hingga 16.777.216 byte; hingga 4 byte per karakter.</p>	VARCHAR(MAX)
<p>TEXT(p)</p> <p>Karakter unicode hingga 65.535 byte; hingga 4 byte per karakter.</p>	Jika p adalah =< 65.535 maka, (p) VARCHAR
<p>TEXT(p)</p> <p>Karakter unicode hingga 16.777.216 byte; hingga 4 byte per karakter.</p>	Jika p adalah => 65.535 dan =< 16,777.216 maka, () VARCHAR MAX
<p>BINARY</p> <p>Karakter single-byte hingga 8.388.608 byte; 1 byte per karakter.</p>	VARCHAR(MAX)
<p>BINARY(p)</p> <p>Karakter single-byte hingga 65.535 byte; 1 byte per karakter.</p>	VARCHAR(p)

Jenis data kepingan salju	Tipe data Amazon Redshift
BINARY(p) Karakter single-byte hingga 8.388.608 byte; 1 byte per karakter.	VARCHAR(MAX)
BOOLEAN	BOOLEAN
DATE	DATE
TIME Nilai waktu antara 00:00:00 dan 23:59:59.999 999999.	VARCHAR(18)
TIME(f) Nilai waktu antara 00:00:00 dan 23:59:59.9 (f).	VARCHAR(n) — 9+dt-attr-1
TIMESTAMP_NTZ	TIMESTAMP
TIMESTAMP_TZ	TIMESTAMPTZ

Pengaturan konversi Snowflake ke Amazon Redshift

Untuk mengedit setelan konversi Snowflake ke Amazon Redshift, pilih Pengaturan AWS SCT di, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Snowflake, lalu pilih Snowflake — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Snowflake ke Amazon Redshift.

Pengaturan konversi Snowflake ke Amazon Redshift AWS SCT di menyertakan opsi untuk hal berikut:

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

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

menambahkan komentar dalam kode yang dikonversi untuk item tindakan dengan tingkat keparahan yang dipilih dan lebih tinggi.

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

- Untuk mengatur jumlah maksimum tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) dalam Panduan Manajemen Amazon Redshift.

AWS SCT mengonversi semua tabel sumber Anda, meskipun ini lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka akan AWS SCT menampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift](#).

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk KEY kolom. Anda dapat memilih opsi ini hanya ketika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Snowflake ke Amazon Redshift

Untuk mengedit setelan pengoptimalan konversi Snowflake ke Amazon Redshift, pilih Pengaturan AWS SCT di, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Snowflake, lalu pilih Snowflake — Amazon Redshift. Di panel kiri, pilih Strategi optimasi. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Snowflake ke Amazon Redshift.

Pengaturan pengoptimalan konversi Snowflake ke Amazon Redshift AWS SCT di menyertakan opsi untuk hal-hal berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses self-tuning di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi optimasi, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Gunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilainya (0-100). Nilai-nilai ini menentukan bobot setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan mempengaruhi pilihan distribusi dan kunci pengurutan. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimasi konversi.

- Untuk Batas kolom kunci Urutkan, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kemunculan nilai yang paling umum dengan jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1-100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih pengguna statistik, pilih pengguna database yang ingin Anda analisis statistik kueri.

Juga, pada tab Strategi optimasi, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menghubungkan ke Gudang Data SQL Server dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Microsoft SQL Server DW ke Amazon Redshift atau Amazon Redshift dan digunakan dalam kombinasi. AWS Glue

Hak istimewa untuk Microsoft SQL Server Data Warehouse sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan gudang data Microsoft SQL Server sebagai sumber:

- VIEW DEFINITION
- VIEW DATABASE STATE
- SELECTPADASHEMA:: *<schema_name>*

Pada contoh sebelumnya, ganti *<source_schema>* placeholder dengan nama sumber `source_schema`.

Ulangi hibah untuk setiap database yang skema Anda konversi.

Selain itu, berikan yang berikut ini, dan jalankan hibah pada database master:

- VIEW SERVER STATE

Batasan untuk Gudang Data SQL Server sebagai sumber

Menggunakan Microsoft SQL Server Parallel Data Warehouse (PDW) sebagai sumber saat ini tidak didukung.

Menghubungkan ke Gudang Data SQL Server sebagai sumber

Gunakan prosedur berikut untuk terhubung ke database sumber Gudang Data SQL Server Anda dengan file AWS Schema Conversion Tool.

Untuk menyambung ke database sumber SQL Server Data Warehouse

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

Parameter	Tindakan
Nama server	Masukkan nama Layanan Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Nama instans	Masukkan nama instance untuk gudang data SQL Server.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada SSLtab:</p> <ul style="list-style-type: none">• Sertifikat server kepercayaan: Pilih opsi ini untuk mempercayai sertifikat server.• Toko kepercayaan: Toko kepercayaan yang Anda atur di pengaturan Global.
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.

Parameter	Tindakan
SQLJalur driver server	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

SQLPengaturan konversi Gudang Data Server ke Amazon Redshift

Untuk mengedit Gudang Data SQL Server ke pengaturan konversi Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Microsoft SQL Server, lalu pilih Microsoft SQL Server — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi SQL Server Data Warehouse ke Amazon Redshift.

SQLPengaturan konversi Gudang Data Server ke Amazon Redshift AWS SCT termasuk opsi untuk hal berikut:

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

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

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

- Untuk mengatur jumlah maksimum tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) dalam Panduan Manajemen Amazon Redshift.

AWS SCT mengonversi semua tabel sumber Anda, meskipun ini lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka akan AWS SCT menampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasikan partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan UNION ALL tampilan dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk tabel sumber tunggal.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasikan setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang menyertakan data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke cluster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift](#).

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk KEY kolom. Anda dapat memilih opsi ini hanya ketika Anda memilih opsi Gunakan pengkodean kompresi.

SQLPengaturan pengoptimalan konversi Gudang Data Server ke Amazon Redshift

Untuk mengedit Gudang Data SQL Server ke pengaturan pengoptimalan konversi Amazon Redshift, pilih Pengaturan di AWS SCT, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Microsoft SQL Server, lalu pilih Microsoft SQL Server — Amazon Redshift. Di panel kiri, pilih Strategi optimasi. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi SQL Server Data Warehouse ke Amazon Redshift.

SQLPengaturan pengoptimalan konversi Gudang Data Server ke Amazon Redshift AWS SCT termasuk opsi untuk hal-hal berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses self-tuning di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi optimasi, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Gunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilainya (0-100). Nilai-nilai ini menentukan bobot setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan

mempengaruhi pilihan distribusi dan kunci pengurutan. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan jumlah baris tabel Maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimasi konversi.

- Untuk Batas kolom kunci Urutkan, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kemunculan nilai yang paling umum dengan jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1-100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih pengguna statistik, pilih pengguna database yang ingin Anda analisis statistik kueri.

Juga, pada tab Strategi optimasi, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menghubungkan ke Gudang Data Teradata dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Teradata ke Amazon Redshift atau Amazon Redshift dan digunakan dalam kombinasi. AWS Glue

Hak istimewa untuk Teradata sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan Teradata sebagai sumber:

- SELECTPADA DBC

- SELECTPADA SYSUDTLIB
- SELECTPADA SYSLIB
- SELECTPADA *<source_database>*
- CREATEPROCEDUREPADA *<source_database>*

Pada contoh sebelumnya, ganti *<source_database>* placeholder dengan nama database sumber.

AWS SCT membutuhkan hak CREATE PROCEDURE istimewa untuk melakukan HELP PROCEDURE terhadap semua prosedur dalam database sumber. AWS SCT tidak menggunakan hak istimewa ini untuk membuat objek baru di database Teradata sumber Anda.

Menghubungkan ke Teradata sebagai sumber

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

Untuk terhubung ke database sumber Teradata

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

Parameter	Tindakan
Nama koneksi	Masukkan nama untuk database Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.
Nama server	Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis Data	Masukkan nama database Teradata.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.
Enkripsi data	Pilih opsi ini untuk mengenkripsi data yang Anda tukarkan dengan database. Jika Anda memilih opsi ini, maka nomor port 443 digunakan untuk mentransfer data terenkripsi antara AWS SCT dan database Teradata Anda.

Parameter	Tindakan
Jalur pengemudi Teradata	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

Menggunakan LDAP otentikasi dengan sumber Teradata

Untuk mengatur otentikasi Protokol Akses Direktori Ringan (LDAP) untuk pengguna Teradata yang menjalankan Microsoft Active Directory di Windows, gunakan prosedur berikut.

Dalam prosedur berikut, domain Active Directory adalah `test.local.com`. Server Windows adalah DC, dan dikonfigurasi dengan pengaturan default. Skrip berikut membuat akun `test_ldap` Active Directory, dan akun ini menggunakan `test_ldap` kata sandi.

Untuk mengatur LDAP otentikasi bagi pengguna Teradata yang menjalankan Microsoft Active Directory di Windows

- Di `/opt/teradata/tdat/tdgss/site` direktori, edit file `TdgssUserConfigFile.xml`. Ubah LDAP bagian menjadi yang berikut.

```
AuthorizationSupported="no"

LdapServerName="DC.test.local.com"
LdapServerPort="389"
LdapServerRealm="test.local.com"
LdapSystemFQDN="dc= test, dc= local, dc=com"
LdapBaseFQDN="dc=test, dc=local, dc=com"
```

- Terapkan perubahan dengan menjalankan konfigurasi sebagai berikut.

```
#cd /opt/teradata/tdgss/bin  
#./run_tdgssconfig
```

3. Uji konfigurasi dengan menjalankan perintah berikut.

```
# /opt/teradata/tdat/tdgss/14.10.03.01/bin/tdsbind -u test_ldap -w test_ldap
```

Output harus serupa dengan yang berikut ini.

```
LdapGroupBaseFQDN: dc=Test, dc=local, dc=com  
LdapUserBaseFQDN: dc=Test, dc=local, dc=com  
LdapSystemFQDN: dc= test, dc= local, dc=com  
LdapServerName: DC.test.local.com  
LdapServerPort: 389  
LdapServerRealm: test.local.com  
LdapClientUseTls: no  
LdapClientTlsReqCert: never  
LdapClientMechanism: SASL/DIGEST-MD5  
LdapServiceBindRequired: no  
LdapClientTlsCRLCheck: none  
LdapAllowUnsafeServerConnect: yes  
UseLdapConfig: no  
AuthorizationSupported: no  
FQDN: CN=test, CN=Users, DC=Anthem, DC=local, DC=com  
AuthUser: ldap://DC.test.local.com:389/CN=test1,CN=Users,DC=test,DC=local,DC=com  
DatabaseName: test  
Service: tdsbind
```

4. Mulai ulang TPA menggunakan perintah berikut.

```
#tpareset -f "use updated TDGSSCONFIG GDO"
```

5. Buat pengguna yang sama di database Teradata seperti di Active Directory, seperti yang ditunjukkan berikut.

```
CREATE USER test_ldap AS PERM=1000, PASSWORD=test_ldap;  
GRANT LOGON ON ALL TO test WITH NULL PASSWORD;
```

Jika Anda mengubah kata sandi pengguna di Active Directory untuk LDAP pengguna Anda, tentukan kata sandi baru ini selama koneksi ke Teradata dalam LDAP mode. Dalam DEFAULT mode, Anda terhubung ke Teradata dengan menggunakan nama LDAP pengguna dan kata sandi apa pun.

Mengkonfigurasi pengumpulan statistik di gudang data Teradata sumber Anda

Untuk mengonversi gudang data Teradata sumber Anda, AWS SCT gunakan statistik untuk mengoptimalkan gudang data Amazon Redshift yang telah dikonversi. Anda dapat mengumpulkan statistik AWS SCT atau mengunggah file statistik. Untuk informasi selengkapnya, lihat [Mengumpulkan atau mengunggah statistik](#).

Untuk memastikan bahwa AWS SCT dapat mengumpulkan statistik dari gudang data Anda, selesaikan tugas prasyarat berikut.

Untuk mengumpulkan statistik dari gudang data Teradata Anda

1. Jalankan kueri berikut untuk mengingat statistik untuk semua tabel di gudang data Anda.

```
collect summary statistics on table_name;
```

Pada contoh sebelumnya, ganti *table_name* dengan nama tabel sumber Anda. Ulangi kueri untuk setiap tabel yang Anda konversi.

2. Jalankan kueri berikut untuk menentukan string akun untuk pengguna, yang Anda gunakan untuk mengonversi gudang data Anda.

```
select * from dbc.accountinfo where username = 'user_name'
```

3. Aktifkan pencatatan kueri untuk pengguna tertentu menggunakan string akun dari contoh sebelumnya.

```
BEGIN QUERY LOGGING WITH OBJECTS, SQL ON ALL ACCOUNT=('M$BUSI$SD$H');
```

Atau, aktifkan pencatatan kueri untuk semua pengguna database.

```
BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT SQLTEXT=0 ON ALL;
```

Setelah Anda menyelesaikan pengumpulan statistik gudang data, matikan pencatatan kueri. Untuk melakukannya, Anda dapat menggunakan contoh kode berikut.

```
end query logging with explain, objects, sql on all account=(' $M$BUSI$$D$H');
```

Mengumpulkan statistik dalam mode offline dari gudang data Teradata sumber Anda

Setelah Anda mengonfigurasi pengumpulan statistik di gudang data Teradata Anda, Anda dapat mengumpulkan statistik dalam proyek Anda AWS SCT . Atau, Anda dapat menggunakan skrip Basic Teradata Query (BTEQ) untuk mengumpulkan statistik dalam mode offline. Kemudian, Anda dapat mengunggah file dengan statistik yang dikumpulkan ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Mengumpulkan atau mengunggah statistik](#).

Untuk mengumpulkan statistik dari gudang data Teradata Anda dalam mode offline

1. Buat `off-line_stats.bteq` skrip dengan konten berikut.

```
.OS IF EXIST column-stats-tera.csv del /F column-stats-tera.csv
.OS IF EXIST table-stats-tera.csv del /F table-stats-tera.csv
.OS IF EXIST column-skew-script-tera.csv del /F column-skew-script-tera.csv
.OS IF EXIST column-skew-stats-tera.csv del /F column-skew-stats-tera.csv
.OS IF EXIST query-stats-tera.csv del /F query-stats-tera.csv
.LOGON your_teradata_server/your_login, your_password
.EXPORT REPORT FILE = table-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

SELECT
  ''' || OREPLACE(COALESCE(c.DatabaseName, ''), '', '""') || ';' ||
  ''' || OREPLACE(COALESCE(c.TableName, ''), '', '""') || ';' ||
  ''' || TRIM(COALESCE(s.reference_count, '0')) || ';' ||
  ''' || TRIM(COALESCE(CAST(p.RowCount AS BIGINT), '0')) || ';' ||
  ''' || CAST(CAST(w.size_in_mb AS DECIMAL (38,1) FORMAT 'Z9.9') AS VARCHAR(38))
  || ';' ||
  ''' || TRIM(COALESCE(r.stat_fk_dep_count, '0')) || ';' ||
  ''' || CAST(CAST(current_timestamp(0) as timestamp(0) format 'YYYY-MM-
DDBHH:MI:SS') as VARCHAR(19)) || '''
(TITLE
  "database_name";"table_name";"reference_count";"row_count";"size_in_mb";"stat_fk_dep_coun
FROM (select databasename, tablename
      from DBC.tablesv
      where tablekind IN ('T','O')
      and databasename = 'your_database_name'
      ) c
left join
```

```

        (select DatabaseName, TableName, max(RowCount) RowCount
         from dbc.tableStatsv
         group by 1,2)p
on p.databasename = c.databasename
and p.tablename = c.tablename
left join
    (SELECT r.ChildDB as DatabaseName,
     r.ChildTable as TableName,
     COUNT(DISTINCT r.ParentTable) reference_count
     FROM DBC.All_RI_ChildrenV r
     GROUP BY r.ChildDB, r.ChildTable) s
on s.databasename = c.databasename
and s.tablename = c.tablename
left join
    (SELECT r.ParentDB as DatabaseName,
     r.ParentTable as TableName,
     COUNT(DISTINCT r.ChildTable) stat_fk_dep_count
     FROM DBC.All_RI_ParentsV r
     GROUP BY r.ParentDB, r.ParentTable) r
on r.databasename = c.databasename
and r.tablename = c.tablename
left join
    (select databasename, tablename,
     sum(currentperm)/1024/1024 as size_in_mb
     from dbc.TableSizeV
     group by 1,2) w
on w.databasename = c.databasename
and w.tablename = c.tablename
WHERE COALESCE(r.stat_fk_dep_count,0) + COALESCE(CAST(p.RowCount AS BIGINT),0) +
  COALESCE(s.reference_count,0) > 0;

.EXPORT RESET

.EXPORT REPORT FILE = column-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000
    ''' || TRIM(COALESCE(CAST(t2.card AS BIGINT), '0')) || ';' ||

SELECT
    ''' || OREPLACE(COALESCE(trim(tv.DatabaseName), ''), '', '""') || ';' ||
    ''' || OREPLACE(COALESCE(trim(tv.TableName), ''), '', '""') || ';' ||
    ''' || OREPLACE(COALESCE(trim(tv.columnname), ''), '', '""') || ';' ||
        ''' || TRIM(COALESCE(CAST(t2.card AS BIGINT), '0')) ||
    ';' ||

```



```

'''' || CAST(current_timestamp AS VARCHAR(19)) || '''' (TITLE
''database_name";"table_name";"column_name";"cardinality";"current_ts''')
FROM dbc.columnsv tv
LEFT JOIN
(
SELECT
c.DatabaseName AS DATABASE_NAME,
c.TABLENAME AS TABLE_NAME,
c.ColumnName AS COLUMN_NAME,
c.UniqueValueCount AS CARD
FROM dbc.tablestatsv c
WHERE c.DatabaseName = 'your_database_name'
AND c.RowCount <> 0
) t2
ON tv.DATABASENAME = t2.DATABASE_NAME
AND tv.TABLENAME = t2.TABLE_NAME
AND tv.COLUMNNAME = t2.COLUMN_NAME
WHERE t2.card > 0;

.EXPORT RESET

.EXPORT REPORT FILE = column-skew-script-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

SELECT
'SELECT CAST(''''' || TRIM(c.DatabaseName) || ''";'''' || TRIM(c.TABLENAME) || ''";''''
|| TRIM(c.COLUMNNAME) || ''";'''' ||
TRIM(CAST(COALESCE(MAX(cnt) * 1.0 / SUM(cnt), 0) AS NUMBER FORMAT '9.9999')) ||
''";'''' ||
CAST(CURRENT_TIMESTAMP(0) AS VARCHAR(19)) || '''''' AS VARCHAR(512))
AS """"DATABASE_NAME""";""TABLE_NAME"";""COLUMN_NAME"";""SKEWED"";""CURRENT_TS""""
FROM(
SELECT COUNT(*) AS cnt
FROM '' || c.DATABASENAME || ''.'' || c.TABLENAME ||
'' GROUP BY '' || c.COLUMNNAME || ''') t' ||
CASE WHEN ROW_NUMBER() OVER(PARTITION BY c.DATABASENAME
ORDER BY c.TABLENAME DESC, c.COLUMNNAME DESC) <> 1
THEN ' UNION ALL '
ELSE ';' END (TITLE '--SKEWED--')
FROM dbc.columnsv c
INNER JOIN
(SELECT databasename, TABLENAME

```

```

FROM dbc.tablesv WHERE tablekind = 'T'
AND databasename = 'your_database_name') t
ON t.databasename = c.databasename
AND t.TABLENAME = c.TABLENAME
INNER JOIN
(SELECT databasename, TABLENAME, columnname FROM dbc.indices GROUP BY 1,2,3
WHERE TRANSLATE_CHK (databasename USING LATIN_TO_UNICODE) + TRANSLATE_CHK
(TABLENAME USING LATIN_TO_UNICODE) + TRANSLATE_CHK (columnname USING
LATIN_TO_UNICODE) = 0
) i
ON i.databasename = c.databasename
AND i.TABLENAME = c.TABLENAME
AND i.columnname = c.columnname
WHERE c.ColumnType NOT IN ('CO','JN','N','++','VA','UT','AN','XM','A1','B0')
ORDER BY c.TABLENAME, c.COLUMNNAME;

.EXPORT RESET

.EXPORT REPORT FILE = column-skew-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 10000

.RUN FILE = column-skew-script-tera.csv

.EXPORT RESET

.EXPORT REPORT FILE = query-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 32000

SELECT
  '' || RTRIM(CAST(SqlTextInfo AS VARCHAR(31900)), ';') || ';' ||
  TRIM(QueryCount) || ';' ||
  TRIM(QueryId) || ';' ||
  TRIM(SqlRowNo) || ';' ||
  TRIM(QueryParts) || ';' ||
  CAST(CURRENT_TIMESTAMP(0) AS VARCHAR(19)) || ''
(TITLE
  "query_text";"query_count";"query_id";"sql_row_no";"query_parts";"current_ts")
FROM
  (
    SELECT QueryId, SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
    SUM(QueryFirstRow) OVER (ORDER BY QueryCount DESC, QueryId ASC, SqlRowNo ASC
    ROWS UNBOUNDED PRECEDING) AS topN

```

```

FROM
(SELECT QueryId,  SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
CASE WHEN
ROW_NUMBER() OVER (PARTITION BY QueryCount, SqlTextInfo ORDER BY QueryId,
SqlRowNo) = 1 AND SqlRowNo = 1
THEN 1 ELSE 0 END AS QueryFirstRow
FROM (
SELECT q.QueryId,  q.SqlTextInfo, q.SqlRowNo,
MAX(q.SqlRowNo) OVER (PARTITION BY q.QueryId) QueryParts,
COUNT(q.SqlTextInfo) OVER (PARTITION BY q.SqlTextInfo) QueryCount
FROM DBC.dbqsqltbl q
INNER JOIN
(
SELECT QueryId
FROM DBC.DBQLogTbl t
WHERE TRIM(t.StatementType) IN ('SELECT')
AND TRIM(t.AbortFlag) = '' AND t.ERRORCODE = 0
AND (CASE WHEN 'All users' IN ('All users') THEN 'All users' ELSE
TRIM(t.USERNAME) END) IN ('All users') --user_name list
AND t.StartTime > CURRENT_TIMESTAMP - INTERVAL '30' DAY
GROUP BY 1
) t
ON q.QueryId = t.QueryId
INNER JOIN
(
SELECT QueryId
FROM DBC.QryLogObjectsV
WHERE ObjectDatabaseName = 'your_database_name'
AND ObjectType = 'Tab'
AND CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
GROUP BY 1
) r
ON r.QueryId = t.QueryId
WHERE q.CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
) t
) t
WHERE SqlTextInfo NOT LIKE '%"%"
) q
WHERE
QueryParts >=1
AND topN <= 50
ORDER BY QueryCount DESC, QueryId, SqlRowNo
QUALIFY COUNT(QueryId) OVER (PARTITION BY QueryId) = QueryParts;

```

```
.EXPORT RESET

.LOGOFF

.QUIT
```

2. Buat `td_run_bteq.bat` file yang menjalankan BTEQ skrip yang Anda buat di langkah sebelumnya. Gunakan konten berikut untuk file ini.

```
@echo off > off-line_stats1.bteq & setLocal enableDELAYedexpansion
@echo off > off-line_stats2.bteq & setLocal enableDELAYedexpansion

set old1=your_teradata_server
set new1=%1
set old2=your_login
set new2=%2
set old3=your_database_name
set new3=%3
set old4=your_password
set /p new4=Input %2 pass?

for /f "tokens=* delims= " %%a in (off-line_stats.bteq) do (
set str1=%%a
set str1=!str1:%old1%=%new1%!
>> off-line_stats1.bteq echo !str1!
)

for /f "tokens=* delims= " %%a in (off-line_stats1.bteq) do (
set str2=%%a
set str2=!str2:%old2%=%new2%!
>> off-line_stats2.bteq echo !str2!
)

type nul > off-line_stats1.bteq

for /f "tokens=* delims= " %%a in (off-line_stats2.bteq) do (
set str3=%%a
set str3=!str3:%old3%=%new3%!
>> off-line_stats1.bteq echo !str3!
)

type nul > off-line_stats2.bteq
```

```
for /f "tokens=* delims= " %%a in (off-line_stats1.bteq) do (
set str4=%%a
set str4=!str4:%old4%=!new4%!
>> off-line_stats2.bteq echo !str4!
)

del .\off-line_stats1.bteq

echo export starting...

bteq -c UTF8 < off-line_stats.bteq > metadata_export.log

pause
```

3. Buat `runme.bat` file yang menjalankan file batch yang Anda buat di langkah sebelumnya. Gunakan konten berikut untuk file ini.

```
.\td_run_bteq.bat ServerName UserName DatabaseName
```

Dalam `runme.bat` file, ganti *ServerName*, *UserName*, dan *DatabaseName* dengan nilai-nilai yang berlaku.

Kemudian, jalankan `runme.bat` file tersebut. Ulangi langkah ini untuk setiap gudang data yang Anda konversi ke Amazon Redshift.

Setelah Anda menjalankan skrip ini, Anda menerima tiga file dengan statistik untuk setiap database. Anda dapat mengunggah file-file ini ke AWS SCT proyek Anda. Untuk melakukannya, pilih gudang data Anda dari panel kiri proyek Anda, dan buka menu konteks (klik kanan). Pilih Unggah Statistik.

Pengaturan konversi Teradata ke Amazon Redshift

Untuk mengedit pengaturan konversi Teradata ke Amazon Redshift, pilih Pengaturan AWS SCT di, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Teradata, lalu pilih Teradata — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Teradata ke Amazon Redshift.

Pengaturan konversi Teradata ke Amazon Redshift AWS SCT di menyertakan opsi untuk hal berikut:

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

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

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

- Untuk mengatur jumlah maksimum tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) dalam Panduan Manajemen Amazon Redshift.

AWS SCT mengonversi semua tabel sumber Anda, meskipun ini lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka akan AWS SCT menampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasikan partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan UNION ALL tampilan dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk tabel sumber tunggal.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasikan setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang menyertakan data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan

ke cluster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift](#).

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk KEY kolom. Anda dapat memilih opsi ini hanya ketika Anda memilih opsi Gunakan pengkodean kompresi.

- Untuk menggunakan daftar kolom eksplisit dalam kode yang dikonversi untuk SELECT * pernyataan, pilih Gunakan deklarasi kolom eksplisit.
- Untuk meniru perilaku kunci primer dan unik di klaster Amazon Redshift Anda, pilih Emulasikan perilaku kunci primer dan unik.

Amazon Redshift tidak menerapkan kunci unik dan utama dan menggunakannya hanya untuk tujuan informasi. Jika Anda menggunakan batasan ini dalam kode Anda, maka pastikan itu AWS SCT mengemulasi perilaku mereka dalam kode yang dikonversi.

- Untuk memastikan keunikan data dalam tabel Amazon Redshift target. Untuk melakukannya, pilih Emulasikan perilaku SET tabel.

Teradata membuat tabel menggunakan elemen SET sintaks sebagai opsi default. Anda tidak dapat menambahkan baris duplikat dalam SET tabel. Jika kode sumber Anda tidak menggunakan kendala keunikan ini, matikan opsi ini. Dalam hal ini, kode yang dikonversi bekerja lebih cepat.

Jika kode sumber Anda menggunakan SET opsi dalam tabel sebagai kendala keunikan, aktifkan opsi ini. Dalam hal ini, AWS SCT menulis ulang INSERT . . SELECT pernyataan dalam kode yang dikonversi untuk meniru perilaku database sumber Anda.

Pengaturan pengoptimalan konversi Teradata ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Teradata ke Amazon Redshift, pilih Pengaturan AWS SCT di, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Teradata, lalu pilih Teradata — Amazon Redshift. Di panel kiri, pilih Strategi optimasi. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Teradata ke Amazon Redshift.

Pengaturan pengoptimalan konversi Teradata ke Amazon Redshift AWS SCT di menyertakan opsi untuk hal berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses self-tuning di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi optimasi, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Gunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilainya (0-100). Nilai-nilai ini menentukan bobot setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan mempengaruhi pilihan distribusi dan kunci pengurutan. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimasi konversi.

- Untuk Batas kolom kunci Urutkan, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kemunculan nilai yang paling umum dengan jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1-100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih pengguna statistik, pilih pengguna database yang ingin Anda analisis statistik kueri.

Juga, pada tab Strategi optimasi, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Menghubungkan database AWS Schema Conversion Tool ke Vertica

Anda dapat menggunakan AWS SCT untuk mengonversi skema, objek kode, dan kode aplikasi dari Vertica ke Amazon Redshift.

Hak istimewa untuk Vertica sebagai sumber

Hak istimewa berikut diperlukan untuk menggunakan Vertica sebagai sumber:

- USAGEPADA SCHEMA *<schema_name>*
- USAGEPADA SCHEMA PUBLIC
- SELECTDI ALL TABLES DALAM SCHEMA *<schema_name>*
- SELECTDI ALL SEQUENCES DALAM SCHEMA *<schema_name>*
- EXECUTEDI ALL FUNCTIONS DALAM SCHEMA *<schema_name>*
- EXECUTEPADA PROCEDURE *<schema_name.procedure_name(procedure_signature)>*

Pada contoh sebelumnya, ganti placeholder sebagai berikut:

- Ganti *schema_name* dengan nama skema sumber.
- Ganti *procedure_name* dengan nama prosedur sumber. Ulangi hibah untuk setiap prosedur yang Anda konversi.
- Ganti *procedure_signature* dengan daftar tipe argumen prosedur yang dibatasi koma.

Menghubungkan ke Vertica sebagai sumber

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

Untuk terhubung ke database sumber Vertica

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

Kotak dialog Tambah sumber muncul.

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

Untuk informasi tentang menggunakan kredensial database dari Secrets Manager, lihat.

[Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#)

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

Parameter	Tindakan
Nama server	Masukkan nama Sistem Nama Domain (DNS) atau alamat IP server basis data sumber Anda.

Parameter	Tindakan
Port server	Masukkan port yang digunakan untuk terhubung ke server database sumber Anda.
Basis Data	Masukkan nama database Vertica.
Nama pengguna dan Kata Sandi	<p>Masukkan kredensi database untuk terhubung ke server basis data sumber Anda.</p> <p>AWS SCT menggunakan kata sandi untuk terhubung ke database sumber Anda hanya ketika Anda memilih untuk terhubung ke database Anda dalam sebuah proyek. Untuk mencegah mengekspos kata sandi untuk basis data sumber Anda, AWS SCT tidak menyimpan kata sandi secara default. Jika Anda menutup AWS SCT proyek Anda dan membukanya kembali, Anda akan diminta kata sandi untuk terhubung ke database sumber Anda sesuai kebutuhan.</p>
Gunakan SSL	<p>Pilih opsi ini untuk menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database Anda. Berikan informasi tambahan berikut, sebagaimana berlaku, pada SSLtab:</p> <ul style="list-style-type: none">• Verifikasi sertifikat server: Pilih opsi ini untuk memverifikasi sertifikat server dengan menggunakan toko kepercayaan.• Toko kepercayaan: Toko kepercayaan yang Anda atur di pengaturan Global.• Key store: Toko kunci yang Anda atur di pengaturan Global.
Menyimpan kata sandi	AWS SCT membuat brankas aman untuk menyimpan SSL sertifikat dan kata sandi database. Dengan mengaktifkan opsi ini, Anda dapat menyimpan kata sandi database dan terhubung dengan cepat ke database tanpa harus memasukkan kata sandi.

Parameter	Tindakan
Jalur pengemudi Vertica	<p>Masukkan jalur ke driver yang akan digunakan untuk terhubung ke database sumber. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool.</p> <p>Jika Anda menyimpan jalur driver di pengaturan proyek global, jalur driver tidak muncul di kotak dialog koneksi. Untuk informasi selengkapnya, lihat Menyimpan jalur driver di pengaturan global.</p>

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

Pengaturan konversi Vertica ke Amazon Redshift

Untuk mengedit pengaturan konversi Vertica ke Amazon Redshift, pilih Pengaturan AWS SCT di, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Vertica, lalu pilih Vertica — Amazon Redshift. AWS SCT menampilkan semua pengaturan yang tersedia untuk konversi Vertica ke Amazon Redshift.

Pengaturan konversi Vertica ke Amazon Redshift AWS SCT di menyertakan opsi untuk hal berikut:

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

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

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

- Untuk mengatur jumlah maksimum tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift target Anda.

Untuk Jumlah maksimum tabel untuk klaster Amazon Redshift target, pilih jumlah tabel yang AWS SCT dapat diterapkan ke cluster Amazon Redshift Anda.

Amazon Redshift memiliki kuota yang membatasi tabel penggunaan untuk jenis node cluster yang berbeda. Jika Anda memilih Otomatis, AWS SCT tentukan jumlah tabel yang akan diterapkan ke klaster Amazon Redshift target Anda tergantung pada jenis node. Secara opsional, pilih nilai secara manual. Untuk informasi selengkapnya, lihat [Kuota dan batasan di Amazon Redshift](#) dalam Panduan Manajemen Amazon Redshift.

AWS SCT mengonversi semua tabel sumber Anda, meskipun ini lebih dari yang dapat disimpan oleh cluster Amazon Redshift Anda. AWS SCT menyimpan kode yang dikonversi dalam proyek Anda dan tidak menerapkannya ke database target. Jika Anda mencapai kuota klaster Amazon Redshift untuk tabel saat Anda menerapkan kode yang dikonversi, maka akan AWS SCT menampilkan pesan peringatan. Selain itu, AWS SCT terapkan tabel ke cluster Amazon Redshift target Anda hingga jumlah tabel mencapai batas.

- Untuk memigrasikan partisi tabel sumber ke tabel terpisah di Amazon Redshift. Untuk melakukannya, pilih Gunakan UNION ALL tampilan dan masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk tabel sumber tunggal.

Amazon Redshift tidak mendukung partisi tabel. Untuk meniru perilaku ini dan membuat kueri berjalan lebih cepat, AWS SCT dapat memigrasikan setiap partisi tabel sumber Anda ke tabel terpisah di Amazon Redshift. Kemudian, AWS SCT buat tampilan yang menyertakan data dari semua tabel ini.

AWS SCT secara otomatis menentukan jumlah partisi dalam tabel sumber Anda. Bergantung pada jenis partisi tabel sumber, angka ini dapat melebihi kuota untuk tabel yang dapat Anda terapkan ke cluster Amazon Redshift Anda. Untuk menghindari mencapai kuota ini, masukkan jumlah maksimum tabel target yang AWS SCT dapat dibuat untuk partisi dari tabel sumber tunggal. Opsi default adalah 368 tabel, yang mewakili partisi selama 366 hari dalam setahun dan dua tabel untuk NO RANGE dan UNKNOWN partisi.

- Untuk menerapkan kompresi ke kolom tabel Amazon Redshift. Untuk melakukannya, pilih Gunakan pengkodean kompresi.

AWS SCT menetapkan pengkodean kompresi ke kolom secara otomatis menggunakan algoritme Amazon Redshift default. Untuk informasi selengkapnya, lihat [Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift](#).

Secara default, Amazon Redshift tidak menerapkan kompresi ke kolom yang didefinisikan sebagai kunci pengurutan dan distribusi. Anda dapat mengubah perilaku ini dan menerapkan kompresi

ke kolom ini. Untuk melakukannya, pilih Gunakan pengkodean kompresi untuk KEY kolom. Anda dapat memilih opsi ini hanya ketika Anda memilih opsi Gunakan pengkodean kompresi.

Pengaturan pengoptimalan konversi Vertica ke Amazon Redshift

Untuk mengedit pengaturan pengoptimalan konversi Vertica ke Amazon Redshift, pilih Pengaturan AWS SCT di, lalu pilih Pengaturan konversi. Dari daftar atas, pilih Vertica, lalu pilih Vertica — Amazon Redshift. Di panel kiri, pilih Strategi optimasi. AWS SCT menampilkan pengaturan pengoptimalan konversi untuk konversi Vertica ke Amazon Redshift.

Pengaturan pengoptimalan konversi Vertica ke Amazon Redshift AWS SCT di menyertakan opsi untuk hal berikut:

- Untuk bekerja dengan optimasi tabel otomatis. Untuk melakukannya, pilih Gunakan penyetelan tabel otomatis Amazon Redshift.

Optimalisasi tabel otomatis adalah proses self-tuning di Amazon Redshift yang secara otomatis mengoptimalkan desain tabel. Untuk informasi selengkapnya, lihat [Bekerja dengan pengoptimalan tabel otomatis](#) di Panduan Pengembang Database Amazon Redshift.

Untuk hanya mengandalkan optimasi tabel otomatis, pilih None for Initial key selection strategy.

- Untuk memilih kunci sortir dan distribusi menggunakan strategi Anda.

Anda dapat memilih kunci sortir dan distribusi menggunakan metadata Amazon Redshift, informasi statistik, atau kedua opsi ini. Untuk strategi pemilihan kunci awal pada tab Strategi optimasi, pilih salah satu opsi berikut:

- Gunakan metadata, abaikan informasi statistik
- Abaikan metadata, gunakan informasi statistik
- Gunakan metadata dan informasi statistik

Bergantung pada opsi yang Anda pilih, Anda dapat memilih strategi pengoptimalan. Kemudian, untuk setiap strategi, masukkan nilainya (0-100). Nilai-nilai ini menentukan bobot setiap strategi. Dengan menggunakan nilai bobot ini, AWS SCT mendefinisikan bagaimana setiap aturan mempengaruhi pilihan distribusi dan kunci pengurutan. Nilai default didasarkan pada praktik terbaik AWS migrasi.

Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam

tabel untuk mendefinisikannya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

- Untuk mengkonfigurasi detail strategi.

Selain menentukan bobot untuk setiap strategi pengoptimalan, Anda dapat mengonfigurasi pengaturan pengoptimalan. Untuk melakukannya, pilih Optimasi konversi.

- Untuk Batas kolom kunci Urutkan, masukkan jumlah kolom maksimum di tombol sortir.
- Untuk nilai ambang miring, masukkan persentase (0-100) dari nilai miring untuk kolom. AWS SCT mengecualikan kolom dengan nilai miring lebih besar dari ambang batas dari daftar kandidat untuk kunci distribusi. AWS SCT mendefinisikan nilai miring untuk kolom sebagai rasio persentase dari jumlah kemunculan nilai yang paling umum dengan jumlah total catatan.
- Untuk kueri N Teratas dari tabel riwayat kueri, masukkan nomor (1-100) kueri yang paling sering digunakan untuk dianalisis.
- Untuk Pilih pengguna statistik, pilih pengguna database yang ingin Anda analisis statistik kueri.

Juga, pada tab Strategi optimasi, Anda dapat menentukan ukuran tabel kecil untuk strategi Temukan tabel kecil. Untuk jumlah baris tabel Min dan Jumlah baris tabel maks, masukkan jumlah baris minimum dan maksimum dalam tabel untuk menganggapnya sebagai tabel kecil. AWS SCT menerapkan gaya ALL distribusi ke tabel kecil. Dalam hal ini, salinan seluruh tabel didistribusikan ke setiap node.

Memetakan tipe data di AWS Schema Conversion Tool

Anda dapat menambahkan beberapa sumber dan basis data target dalam satu AWS SCT proyek. Melakukan hal ini menyederhanakan pengelolaan proyek, ketika Anda memigrasikan beberapa database ke platform target yang berbeda.

Setelah Anda membuat proyek baru dan menambahkan basis data sumber dan target, buat aturan pemetaan. AWS SCT memerlukan setidaknya satu aturan pemetaan untuk membuat laporan penilaian migrasi dan mengonversi skema database.

Aturan pemetaan menggambarkan pasangan sumber-target yang mencakup skema database sumber atau database sumber dan platform basis data target. Anda dapat membuat beberapa aturan pemetaan dalam satu AWS SCT proyek. Gunakan aturan pemetaan untuk mengonversi setiap skema basis data sumber ke platform basis data target yang tepat.

Untuk mengubah nama skema Anda dalam kode yang dikonversi, siapkan aturan migrasi. Misalnya, dengan aturan migrasi, Anda dapat mengganti nama skema, menambahkan awalan ke nama objek, mengubah pemeriksaan kolom, atau mengubah tipe data. Untuk menerapkan perubahan ini ke kode yang dikonversi, pastikan Anda membuat aturan migrasi sebelum mengonversi skema sumber. Untuk informasi selengkapnya, lihat [Menerapkan aturan migrasi](#).

Anda dapat membuat aturan pemetaan hanya untuk pasangan konversi database yang didukung. Untuk daftar pasangan konversi yang didukung, lihat [Menghubungkan ke database sumber dengan AWS Schema Conversion Tool](#).

Jika Anda membuka proyek yang disimpan dalam AWS SCT versi 1.0.655 atau sebelumnya, AWS SCT secara otomatis membuat aturan pemetaan untuk semua skema basis data sumber ke platform basis data target. Untuk menambahkan platform basis data target lainnya, hapus aturan pemetaan yang ada dan kemudian buat aturan pemetaan baru.

Topik

- [Memetakan tipe data baru di AWS Schema Conversion Tool](#)
- [Mengedit pemetaan tipe data di AWS Schema Conversion Tool](#)
- [Pemetaan ke target virtual di AWS Schema Conversion Tool](#)
- [Keterbatasan pemetaan tipe data di AWS Schema Conversion Tool](#)

Memetakan tipe data baru di AWS Schema Conversion Tool

Anda dapat membuat beberapa aturan pemetaan dalam satu proyek. AWS SCT menyimpan aturan pemetaan sebagai bagian dari proyek Anda. Dengan proyek Anda terbuka, gunakan prosedur berikut untuk menambahkan aturan pemetaan baru.

Untuk membuat aturan pemetaan

1. Pada menu Tampilan, pilih Tampilan pemetaan.
2. Di panel kiri, pilih skema atau database untuk ditambahkan ke aturan pemetaan.
3. Di panel kanan, pilih platform database target untuk skema sumber atau database yang dipilih.

Anda dapat memilih platform database virtual sebagai target. Untuk informasi selengkapnya, lihat [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).

4. Pilih Buat pemetaan.

AWS SCT menambahkan aturan pemetaan baru ini ke daftar pemetaan Server.

Tambahkan aturan pemetaan untuk semua pasangan konversi. Untuk membuat laporan penilaian atau mengonversi skema database, pilih Tampilan utama pada menu Tampilan.

AWS SCT menyoroti dengan huruf tebal semua objek skema yang merupakan bagian dari aturan pemetaan.

Mengedit pemetaan tipe data di AWS Schema Conversion Tool

Anda dapat memfilter atau menghapus aturan pemetaan yang ada, dan menambahkan aturan pemetaan baru di proyek AWS Schema Conversion Tool (AWS SCT) Anda.

Saat Anda membuat aturan pemetaan untuk seluruh basis data sumber, AWS SCT buat satu aturan pemetaan untuk setiap skema basis data sumber. Untuk proyek yang melibatkan lusinan skema atau bahkan database, mungkin sulit dipahami, target mana yang digunakan untuk skema tertentu. Untuk menemukan aturan pemetaan skema Anda dengan cepat, gunakan satu atau beberapa opsi filter berikut. AWS SCT

Untuk memfilter aturan pemetaan

1. Pada menu Tampilan, pilih Tampilan pemetaan.

2. Untuk server Sumber, pilih database sumber.

Filter default adalah Semua, yang berarti AWS SCT menampilkan aturan pemetaan untuk semua database sumber.

3. Untuk skema Sumber, masukkan nama skema sumber. Gunakan percent (%) sebagai wildcard untuk mengganti sejumlah simbol apa pun dalam nama skema.

Filter default adalah wildcard%, yang berarti AWS SCT menampilkan aturan pemetaan untuk semua nama skema database sumber.

4. Untuk aturan migrasi Has, pilih Ya untuk menampilkan aturan pemetaan yang membuat aturan migrasi data. Pilih Tidak untuk menampilkan aturan pemetaan yang tidak memiliki aturan migrasi data. Untuk informasi selengkapnya, lihat [Membuat aturan migrasi data di AWS SCT](#).

Filter default adalah Semua, yang berarti AWS SCT menampilkan semua aturan pemetaan.

5. Untuk server Target, pilih database target.

Filter default adalah Semua, yang berarti AWS SCT menampilkan aturan pemetaan untuk semua database target.

Dengan proyek Anda terbuka, gunakan prosedur berikut untuk menghapus aturan pemetaan. Untuk informasi selengkapnya tentang menambahkan aturan pemetaan, lihat [Memetakan tipe data baru di AWS Schema Conversion Tool](#).

Untuk menghapus aturan pemetaan

1. Pada menu Tampilan, pilih Tampilan pemetaan.
2. Untuk pemetaan Server, pilih aturan pemetaan yang akan dihapus.
3. Pilih Hapus pemetaan yang dipilih.

AWS SCT menghapus aturan pemetaan yang dipilih.

Pemetaan ke target virtual di AWS Schema Conversion Tool

Anda dapat melihat bagaimana AWS SCT mengubah skema basis data sumber Anda ke platform basis data target yang didukung. Untuk melakukannya, Anda tidak perlu terhubung ke database target yang ada. Sebagai gantinya, Anda dapat memilih platform database target virtual di panel kanan saat Anda membuat aturan pemetaan. Untuk informasi selengkapnya, lihat [Memetakan tipe](#)

[data baru di AWS Schema Conversion Tool](#). Pastikan Anda memperluas Server, Tidak ada SQL cluster, dan ETLnode di panel kanan untuk melihat daftar platform database target virtual.

AWS SCT mendukung platform basis data target virtual berikut:

- Amazon Aurora Edisi Kompatibel Saya SQL
- Amazon Aurora SQL Postgre -Edisi Kompatibel
- Amazon DynamoDB
- Amazon Redshift
- Amazon Redshift dan AWS Glue
- AWS Glue
- AWS Glue Studio
- Babelfish untuk Aurora Postgre SQL
- MariaDB
- SQLServer Microsoft
- Saya SQL
- Oracle
- Postgre SQL

Jika Anda menggunakan Babelfish untuk Aurora Postgre SQL sebagai platform database target, Anda hanya dapat membuat laporan penilaian migrasi database. Untuk informasi selengkapnya, lihat [the section called “Laporan penilaian”](#).

Jika Anda menggunakan platform basis data target virtual, Anda dapat menyimpan kode yang dikonversi ke file. Untuk informasi selengkapnya, lihat [the section called “Menyimpan skema konversi Anda”](#).

Keterbatasan pemetaan tipe data di AWS Schema Conversion Tool

Batasan berikut berlaku saat mengonversi skema menggunakan beberapa server dalam satu AWS SCT proyek:

- Anda dapat menambahkan server yang sama ke proyek hanya sekali.
- Anda tidak dapat memetakan skema server ke skema target tertentu, hanya ke server target. AWS SCT membuat skema target selama konversi.

- Anda tidak dapat memetakan objek sumber tingkat rendah ke server target.
- Anda dapat memetakan satu skema sumber ke hanya satu server target dalam sebuah proyek.
- Pastikan untuk memetakan sumber ke server target untuk membuat laporan penilaian, mengonversi skema, atau mengekstrak data.

Bekerja dengan laporan di AWS Schema Conversion Tool

Ketika Anda merencanakan konversi database, akan sangat membantu untuk membuat beberapa laporan untuk membantu Anda memahami apa yang terlibat. Anda dapat membuat laporan menggunakan AWS Schema Conversion Tool.

Anda dapat menggunakan AWS SCT untuk membuat laporan penilaian migrasi database. Dengan laporan ini, Anda mendapatkan ringkasan tugas konversi skema dan detail untuk item yang tidak dapat secara otomatis dikonversi ke database target Anda. Anda dapat menggunakan laporan ini untuk mengevaluasi seberapa banyak proyek dapat diselesaikan dengan menggunakan AWS SCT, dan apa lagi yang Anda butuhkan untuk menyelesaikan konversi. Untuk membuat laporan penilaian, gunakan Buat Laporan dari menu konteks (klik kanan) database di AWS SCT.

Topik

- [Menggunakan laporan penilaian di AWS Schema Conversion Tool](#)

Menggunakan laporan penilaian di AWS Schema Conversion Tool

Bagian penting dari laporan AWS Schema Conversion Tool ini adalah laporan penilaian yang dihasilkannya untuk memperkirakan kompleksitas konversi skema Anda. Laporan penilaian migrasi database ini merangkum semua tugas konversi skema dan merinci item tindakan untuk skema yang tidak dapat dikonversi ke mesin DB instans DB target Anda. Anda dapat melihat laporan dalam aplikasi atau mengeksportnya sebagai nilai (CSV) atau PDF file yang dipisahkan koma.

Jika Anda menambahkan beberapa sumber dan basis data target dalam satu proyek, AWS SCT agregat laporan untuk semua pasangan konversi ke dalam satu laporan penilaian migrasi database.

Anda dapat menggunakan platform basis data target virtual untuk menghasilkan laporan penilaian dan memahami kompleksitas migrasi ke platform database yang dipilih. Dalam hal ini, Anda tidak perlu terhubung ke platform basis data target Anda. Misalnya, Anda dapat menggunakan Babelfish untuk Aurora Postgre SQL sebagai platform database target virtual untuk membuat laporan penilaian migrasi database. Untuk informasi selengkapnya tentang platform basis data target virtual, lihat [the section called “Pemetaan target virtual”](#).

Laporan penilaian migrasi mencakup hal-hal berikut:

- Ringkasan eksekutif

- Evaluasi lisensi
- Dukungan cloud, menunjukkan fitur apa pun dalam basis data sumber yang tidak tersedia pada target.
- Rekomendasi, termasuk konversi objek server, saran cadangan, dan perubahan server tertaut

Laporan ini juga mencakup perkiraan jumlah upaya yang diperlukan untuk menulis kode yang setara untuk instans DB target Anda yang tidak dapat dikonversi secara otomatis.

Jika Anda menggunakannya AWS SCT untuk memigrasikan skema yang ada ke instans Amazon RDS DB, Anda dapat menggunakan laporan tersebut untuk membantu menganalisis persyaratan untuk pindah ke AWS Cloud dan mengubah jenis lisensi Anda.

Topik

- [Membuat laporan penilaian di AWS Schema Conversion Tool](#)
- [Melihat laporan penilaian di AWS Schema Conversion Tool](#)
- [Menyimpan laporan penilaian di AWS Schema Conversion Tool](#)
- [Mengkonfigurasi laporan penilaian di AWS Schema Conversion Tool](#)
- [Membuat laporan penilaian multiserver di AWS Schema Conversion Tool](#)

Membuat laporan penilaian di AWS Schema Conversion Tool

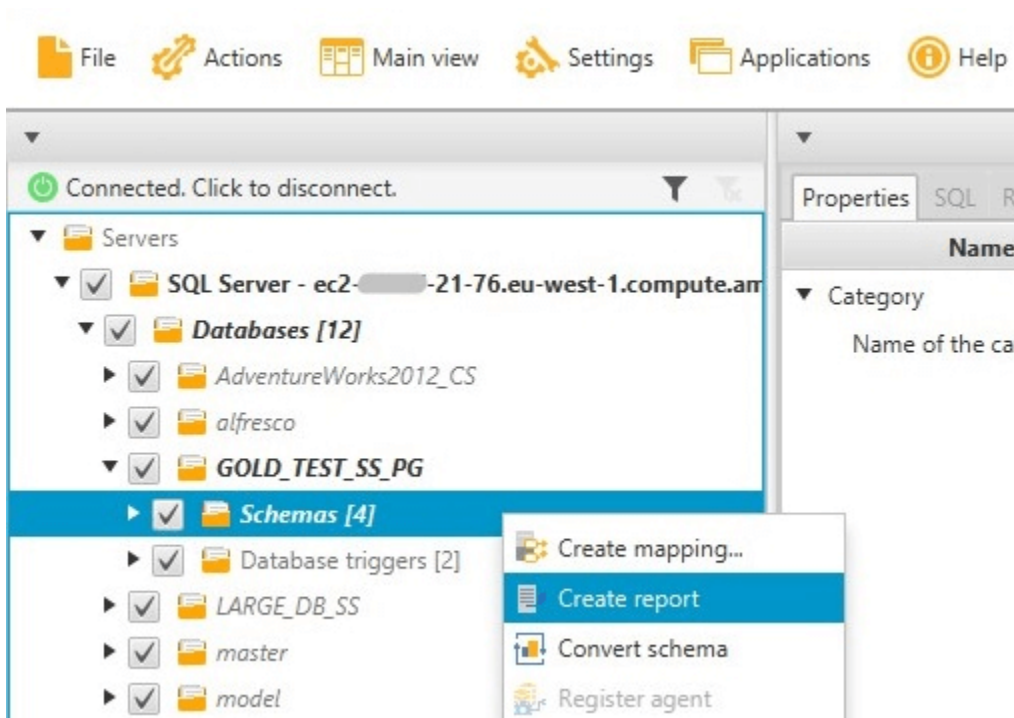
Gunakan prosedur berikut untuk membuat laporan penilaian migrasi database.

Untuk membuat laporan penilaian migrasi database

1. Pastikan Anda membuat aturan pemetaan untuk skema database sumber untuk membuat laporan penilaian. Untuk informasi selengkapnya, lihat [Memetakan tipe data baru di AWS Schema Conversion Tool](#).
2. Pada menu Tampilan, pilih Tampilan utama.
3. Di panel kiri yang menampilkan skema basis data sumber Anda, pilih objek skema untuk membuat laporan penilaian. Untuk menyertakan beberapa skema database ke dalam laporan, pilih node induk, misalnya Skema.

Pastikan Anda memilih kotak centang untuk semua objek skema untuk membuat laporan penilaian.

4. Buka menu konteks (klik kanan) untuk objek, lalu pilih Buat laporan.



Melihat laporan penilaian di AWS Schema Conversion Tool

Setelah Anda membuat laporan penilaian, tampilan laporan penilaian akan terbuka, menampilkan tab berikut:

- Ringkasan
- Item Tindakan

Tab Ringkasan menunjukkan item yang dikonversi secara otomatis atau tidak dikonversi.

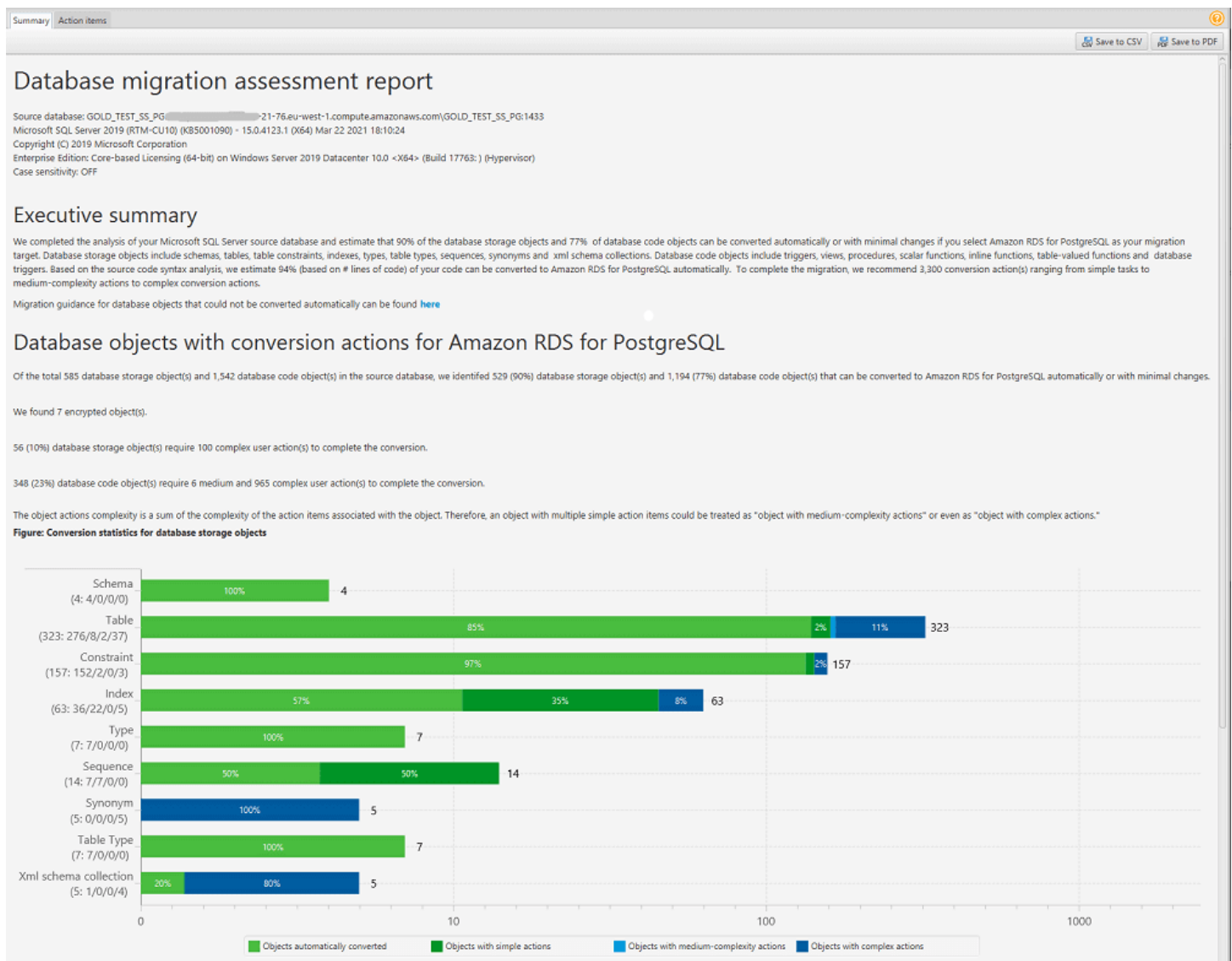
Tab Item Tindakan menunjukkan item yang tidak dapat dikonversi secara otomatis, dan rekomendasi tentang apa yang harus dilakukan terhadapnya.

Topik

- [Ringkasan laporan penilaian](#)
- [Item tindakan laporan penilaian](#)
- [Pesan peringatan laporan penilaian](#)

Ringkasan laporan penilaian

Tab Ringkasan menampilkan informasi ringkasan dari laporan penilaian migrasi database. Ini menunjukkan item yang dikonversi secara otomatis, dan item yang tidak dikonversi secara otomatis.



Untuk item skema yang tidak dapat dikonversi secara otomatis ke mesin basis data target, ringkasan mencakup perkiraan upaya yang diperlukan untuk membuat item skema dalam instans DB target Anda yang setara dengan yang ada di sumber Anda.

Laporan tersebut mengkategorikan perkiraan waktu untuk mengonversi item skema ini sebagai berikut:

- Sederhana — Tindakan yang dapat diselesaikan dalam waktu kurang dari dua jam.
- Medium — Tindakan yang lebih kompleks dan dapat diselesaikan dalam dua hingga enam jam.

- **Signifikan** — Tindakan yang sangat kompleks dan membutuhkan waktu lebih dari enam jam untuk menyelesaikannya.

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

License evaluation

Our analysis shows that current schema uses the following Enterprise Edition features unavailable in Standard Edition.

Feature	Description
Database In-Memory	Oracle Database In-Memory optimizes both analytics and mixed workload OLTP, delivering outstanding performance for transactions while simultaneously supporting real-time analytics, business intelligence, and reports.
Materialized View Query Rewrite	Oracle Database employs an extremely powerful process called query rewrite to quickly answer the query using materialized views.
Partitioning	Partitioning is powerful functionality that allows tables, indexes, and index-organized tables to be subdivided into smaller pieces, enabling these database objects to be managed and accessed at a finer level of granularity.
Oracle Advanced Security/TDE	Oracle Advanced Security provides two important preventive controls to protect sensitive data at the source: encryption and redaction. Together, these two controls form the foundation of Oracle's defense-in-depth, multi-layered database security solution.

If you choose Standard Edition as your migration target, remove dependencies on these features.

Cloud support

Our analysis shows that your current schema uses the following features that require configuration steps in Amazon RDS for Oracle.

Feature	Description
Locator	Oracle Locator provides capabilities that are typically required to support internet and wireless service-based applications and partner-based GIS solutions. Oracle Locator is a limited subset of Oracle Spatial. Please read prerequisites and configuration steps in the next article: Oracle Locator .
Spatial	Oracle Spatial provides a SQL schema and functions that facilitate the storage, retrieval, update, and query of collections of spatial data in an Oracle database. Please read prerequisites and configuration steps in the next article: Oracle Spatial .
Oracle XML DB	Oracle XML DB provides full support for all of the key XML standards, including XML Namespaces, DOM, XQuery, SQL/XML and XSLT. Amazon RDS for Oracle supports XML DB feature without the XML DB Protocol Server. Please read prerequisites and configuration steps in the next article: Oracle XML DB option .

If choose Amazon RDS for Oracle as your migration target, please follow the abovementioned steps to continue to use these features on the target database after migration completes.

Item tindakan laporan penilaian

Tampilan laporan penilaian juga menyertakan tab Item Tindakan. Tab ini berisi daftar item yang tidak dapat dikonversi secara otomatis ke mesin database instans Amazon RDS DB target Anda. Jika Anda memilih item tindakan dari daftar, AWS SCT sorot item dari skema Anda yang berlaku untuk item tindakan.

Laporan ini juga berisi rekomendasi tentang cara mengonversi item skema secara manual. Misalnya, setelah penilaian berjalan, laporan terperinci untuk database/skema menunjukkan kepada Anda upaya yang diperlukan untuk merancang dan mengimplementasikan rekomendasi untuk mengonversi item Tindakan. Untuk informasi selengkapnya tentang memutuskan cara menangani konversi manual, lihat [Mengkonversi skema usinf AWS SCT](#).

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the source database structure, including servers, databases, tables, and schemas. The main pane shows a list of issues with their recommended actions and occurrence counts. Below the issues, a SQL procedure definition is shown for the target Amazon RDS MySQL category.

Issues:

- Issue: 609: MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required**
Recommended action: Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table. After the INSERT operation, you can make use of the rows saved in the temporary table.
Number of occurrences: 1 | Documentation reference(s): <http://dev.mysql.com/doc/refman/8.0/en/insert.html>
- Issue: 681: MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically**
Recommended action: Use non-clustered indexes.
Number of occurrences: 2
- Issue: 794: MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype**
Recommended action: Please review generated code and modify it if necessary.
Number of occurrences: 1
Parameter: @InputPosNo (Number of occurrences: 1)
MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype
- Issue: 826: Check the default value for a DateTime variable**
Recommended action: Check the default value for a DateTime variable.
Number of occurrences: 1
- Issue: 844: MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision**
Recommended action: Review your transformed code and modify it if necessary to avoid a loss of accuracy.
Number of occurrences: 8 | Documentation reference(s): <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
- Issue: 9997: Unable to resolve objects**
Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually.
Number of occurrences: 3
- Issue: 690: MySQL doesn't support table types**
Recommended action: Perform a manual conversion.
Number of occurrences: 1
- Issue: 811: Unable to convert functions**
Recommended action: Create a user-defined function.
Number of occurrences: 12

SQL Procedure Definition:

```

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

```

Pesan peringatan laporan penilaian

Untuk menilai kompleksitas konversi ke mesin database lain, AWS SCT memerlukan akses ke objek dalam database sumber Anda. Ketika tidak SCT dapat melakukan penilaian karena masalah ditemui selama pemindaian, pesan peringatan dikeluarkan yang menunjukkan persentase konversi keseluruhan berkurang.

Warning!

We found that your source database may be configured not in correct way or you have not enough privileges for reading all necessary metadata. Please check your configuration and run report again. For more details please review [help documentation](#).

List of Action Items to review:

- Issue 9997** Unable to resolve objects (number of occurrences: 3)
Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually.

Berikut adalah alasan mengapa AWS SCT mungkin mengalami masalah selama pemindaian:

- Akun pengguna yang terhubung ke database tidak memiliki akses ke semua objek yang diperlukan.
- Objek yang dikutip dalam skema tidak lagi ada dalam database.

- SCT sedang mencoba menilai objek yang dienkripsi.

Untuk informasi selengkapnya tentang izin keamanan yang SCT diperlukan dan hak istimewa untuk database Anda, lihat [Menghubungkan ke database sumber dengan AWS Schema Conversion Tool](#) bagian basis data sumber yang sesuai dalam panduan ini.

Menyimpan laporan penilaian di AWS Schema Conversion Tool

Setelah [membuat laporan penilaian migrasi database](#), Anda dapat menyimpan salinan lokal laporan penilaian migrasi database sebagai PDF file atau file value (CSV) yang dipisahkan koma.

Untuk menyimpan laporan penilaian migrasi database sebagai PDF file

1. Di menu atas, pilih Lihat, lalu pilih Tampilan laporan penilaian.
2. Pilih tab Ringkasan.
3. Pilih Simpan ke PDF kanan atas.

Untuk menyimpan laporan penilaian migrasi database sebagai CSV file

1. Di menu atas, pilih Lihat, lalu pilih Tampilan laporan penilaian.
2. Pilih tab Ringkasan.
3. Pilih Simpan ke CSV kanan atas.

PDF file berisi informasi ringkasan dan item tindakan, seperti yang ditunjukkan pada contoh berikut.

Database objects with conversion actions for Amazon RDS for PostgreSQL

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

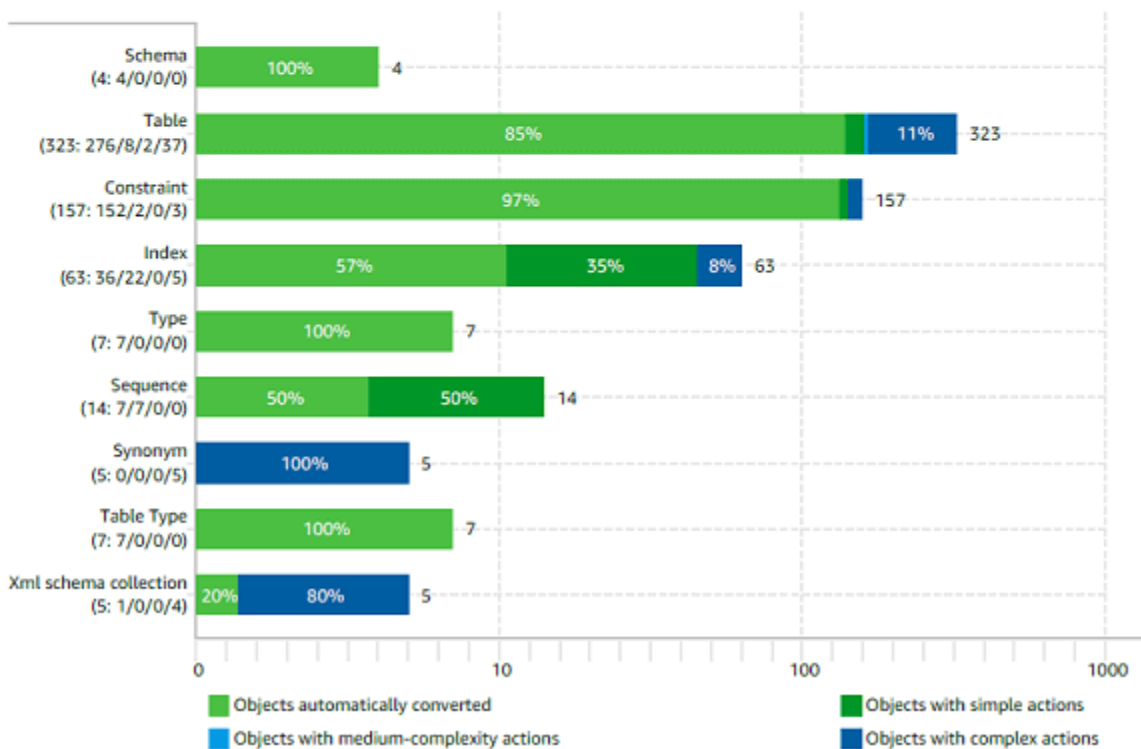
We found 7 encrypted object(s).

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

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

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

Figure: Conversion statistics for database storage objects



Saat Anda memilih CSV opsi Simpan ke, AWS SCT buat tiga CSV file.

CSVFile pertama berisi informasi berikut tentang item tindakan:

- Kategori
- Keberadaan - Nama file, nomor baris, dan posisi item
- Nomor item tindakan
- Subjek
- Grup

- Deskripsi
- Referensi dokumentasi
- Tindakan yang disarankan
- Perkiraan kompleksitas

CSVFile kedua menyertakan `Action_Items_Summary` akhiran dalam namanya dan berisi informasi tentang jumlah kemunculan semua item tindakan.

Dalam contoh berikut, nilai dalam kolom Upaya kurva pembelajaran menunjukkan jumlah upaya yang diperlukan untuk merancang pendekatan untuk mengonversi setiap item tindakan. Nilai dalam Upaya untuk mengonversi kemunculan kolom item tindakan menunjukkan upaya yang diperlukan untuk mengonversi setiap item tindakan, mengikuti pendekatan yang dirancang. Nilai-nilai yang digunakan untuk menunjukkan tingkat usaha yang dibutuhkan didasarkan pada skala tertimbang, mulai dari rendah (terkecil) hingga tinggi (kebanyakan).

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

CSVFile ketiga termasuk `Summary` dalam namanya dan berisi ringkasan berikut:

- Kategori
- Jumlah objek
- Objek secara otomatis dikonversi
- Objek dengan tindakan sederhana
- Objek dengan tindakan kompleksitas sedang
- Objek dengan tindakan kompleks
- Total baris kode

Mengkonfigurasi laporan penilaian di AWS Schema Conversion Tool

Anda dapat mengonfigurasi jumlah detail yang AWS SCT disertakan ke dalam laporan penilaian.

Untuk mengonfigurasi laporan penilaian migrasi database

1. Pada menu Pengaturan, pilih Pengaturan global, lalu pilih Laporan penilaian.
2. Untuk kemunculan item Tindakan, pilih Lima masalah pertama hanya untuk membatasi jumlah item tindakan dari satu jenis dalam laporan penilaian. Pilih Semua masalah untuk menyertakan semua item tindakan dari setiap jenis dalam laporan penilaian.
3. Untuk file yang dianalisis SQL skrip, pilih Daftar tidak lebih dari **X** file untuk membatasi jumlah file SQL skrip dalam laporan penilaian **X**. Masukkan jumlah file. Pilih Daftar semua file yang dianalisis untuk menyertakan semua file SQL skrip dalam laporan penilaian.
4. Pilih Buka laporan setelah menyimpan untuk membuka file secara otomatis setelah Anda menyimpan salinan lokal laporan penilaian migrasi database. Untuk informasi selengkapnya, lihat

Setelah [membuat laporan penilaian migrasi database](#), Anda dapat menyimpan salinan lokal laporan penilaian migrasi database sebagai PDF file atau file value (CSV) yang dipisahkan koma.

Untuk menyimpan laporan penilaian migrasi database sebagai PDF file

1. Di menu atas, pilih Lihat, lalu pilih Tampilan laporan penilaian.
 2. Pilih tab Ringkasan.
 3. Pilih Simpan ke PDF kanan atas.
-

Untuk menyimpan laporan penilaian migrasi database sebagai CSV file

1. Di menu atas, pilih Lihat, lalu pilih Tampilan laporan penilaian.
 2. Pilih tab Ringkasan.
 3. Pilih Simpan ke CSV kanan atas.
-

PDFFile berisi informasi ringkasan dan item tindakan, seperti yang ditunjukkan pada contoh berikut.

Database objects with conversion actions for Amazon RDS for PostgreSQL

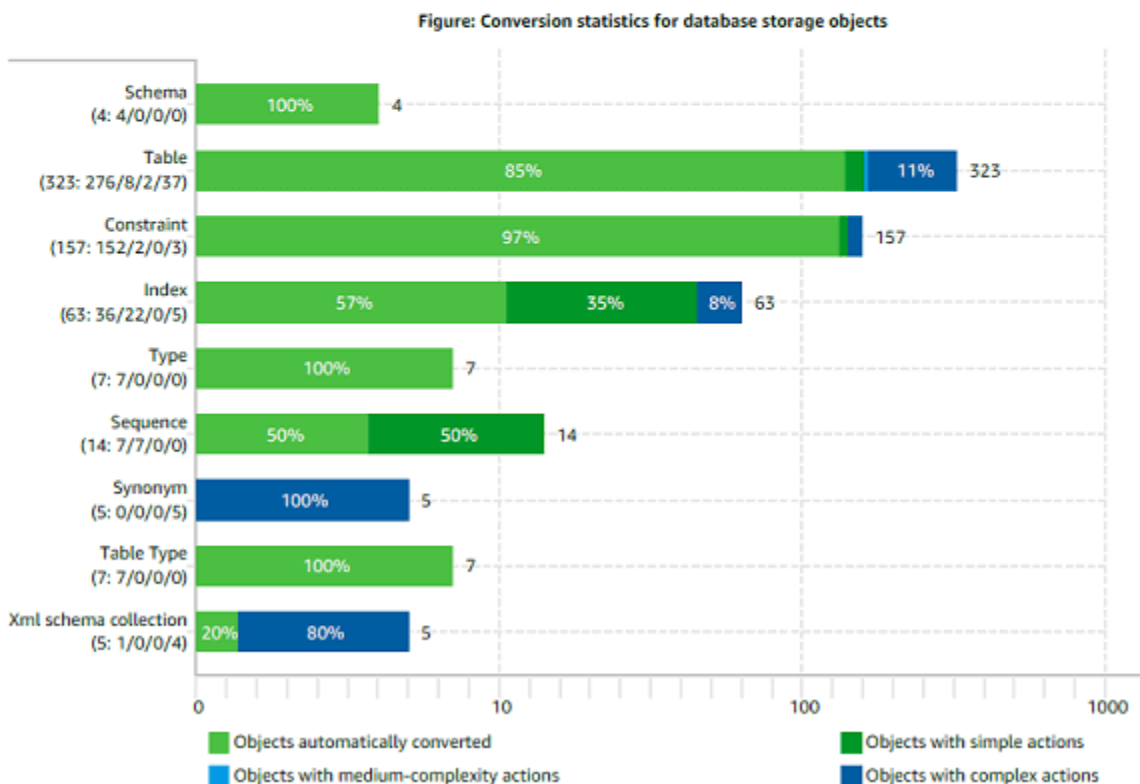
Of the total 585 database storage object(s) and 1,542 database code object(s) in the source database, we identified 529 (90%) database storage object(s) and 1,194 (77%) database code object(s) that can be converted to Amazon RDS for PostgreSQL automatically or with minimal changes.

We found 7 encrypted object(s).

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

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

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



Saat Anda memilih CSV opsi Simpan ke, AWS SCT buat tiga CSV file.

CSVFile pertama berisi informasi berikut tentang item tindakan:

- Kategori
- Keberadaan - Nama file, nomor baris, dan posisi item
- Nomor item tindakan
- Subjek
- Grup

- Deskripsi

- Referensi dokumentasi

- Tindakan yang disarankan

- Perkiraan kompleksitas

CSVFile kedua menyertakan `Action_Items_Summary` akhiran dalam namanya dan berisi informasi tentang jumlah kemunculan semua item tindakan.

Dalam contoh berikut, nilai dalam kolom Upaya kurva pembelajaran menunjukkan jumlah upaya yang diperlukan untuk merancang pendekatan untuk mengonversi setiap item tindakan. Nilai dalam Upaya untuk mengonversi kemunculan kolom item tindakan menunjukkan upaya yang diperlukan untuk mengonversi setiap item tindakan, mengikuti pendekatan yang dirancang. Nilai-nilai yang digunakan untuk menunjukkan tingkat usaha yang dibutuhkan didasarkan pada skala tertimbang, mulai dari rendah (terkecil) hingga tinggi (kebanyakan).

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

CSVFile ketiga termasuk `Summary` dalam namanya dan berisi ringkasan berikut:

- Kategori

- Jumlah objek

- Objek secara otomatis dikonversi

- Objek dengan tindakan sederhana

- Objek dengan tindakan kompleksitas sedang

- Objek dengan tindakan kompleks

- Total baris kode

Membuat laporan penilaian multiserver di AWS Schema Conversion Tool

Untuk menentukan arah target terbaik untuk lingkungan Anda secara keseluruhan, buat laporan penilaian multiserver.

Laporan penilaian multiserver mengevaluasi beberapa server berdasarkan masukan yang Anda berikan untuk setiap definisi skema yang ingin Anda nilai. Definisi skema Anda berisi parameter koneksi server database dan nama lengkap setiap skema. Setelah menilai setiap skema, buat AWS SCT ringkasan, laporan penilaian agregat untuk migrasi database di beberapa server Anda. Laporan ini menunjukkan perkiraan kompleksitas untuk setiap target migrasi yang mungkin.

Anda dapat menggunakan AWS SCT untuk membuat laporan penilaian multiserver untuk database sumber dan target berikut.

Basis data sumber	Basis data target
Amazon Redshift	Amazon Redshift
Database Azure SQL	Aurora Saya, Aurora SQL Postgre, Saya, Postgre SQL SQL SQL
Analisis Sinaps Azure	Amazon Redshift
BigQuery	Amazon Redshift
Greenplum	Amazon Redshift
IBMDb2 untuk z/OS	Amazon Aurora My SQL -Compatible Edition (Aurora My), SQL Amazon Aurora Postgre SQL -Edisi Kompatibel (Aurora Postgre), My, Postgre SQL SQL SQL
IBMDb2 LUW	Aurora Saya, Aurora Postgre, SQL MariaDB, Saya, SQL Postgre SQL SQL
SQLServer Microsoft	Aurora My, SQL Aurora PostgreSQL, Amazon Redshift, Babelfish untuk Aurora Postgre, MariaDB, Microsoft Server, My, Postgre SQL SQL SQL SQL

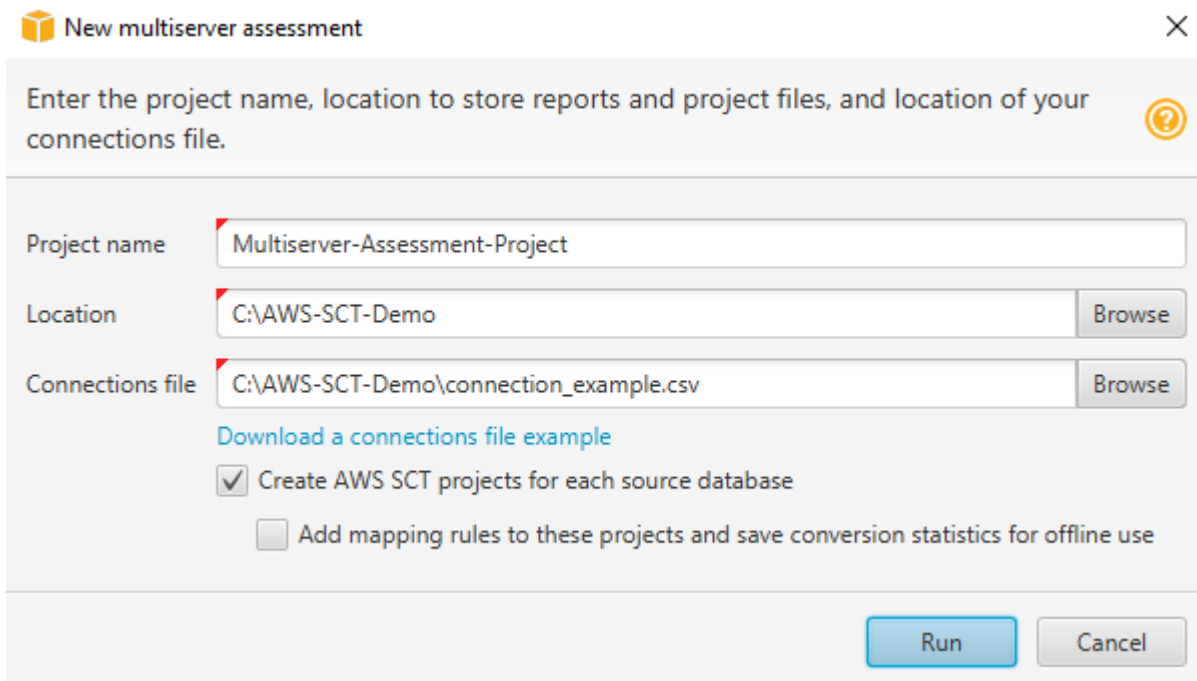
Basis data sumber	Basis data target
Saya SQL	Aurora Postgre, Saya, Postgre SQL SQL SQL
Netezza	Amazon Redshift
Oracle	Aurora Saya, SQL Aurora PostgreSQL, Amazon Redshift, MariaDB, Saya, Oracle, Postgre SQL SQL
Postgre SQL	Aurora Saya, Aurora SQL Postgre, Saya, Postgre SQL SQL SQL
SAP ASE	Aurora Saya, Aurora Postgre, SQL MariaDB, Saya, SQL Postgre SQL SQL
Kepingan salju	Amazon Redshift
Teradata	Amazon Redshift
Vertica	Amazon Redshift

Melakukan penilaian multiserver

Gunakan prosedur berikut untuk melakukan penilaian multiserver dengan AWS SCT. Anda tidak perlu membuat proyek baru AWS SCT untuk melakukan penilaian multiserver. Sebelum memulai, pastikan Anda telah menyiapkan file value (CSV) dipisahkan koma dengan parameter koneksi database. Juga, pastikan bahwa Anda telah menginstal semua driver database yang diperlukan dan mengatur lokasi driver dalam AWS SCT pengaturan. Untuk informasi selengkapnya, lihat [Menginstal JDBC driver untuk AWS Schema Conversion Tool](#).

Untuk melakukan penilaian multiserver dan membuat laporan ringkasan agregat

1. Di AWS SCT, pilih File, Penilaian multiserver baru. Kotak dialog penilaian multiserver baru terbuka.



New multiserver assessment

Enter the project name, location to store reports and project files, and location of your connections file.

Project name: Multiserver-Assessment-Project

Location: C:\AWS-SCT-Demo [Browse]

Connections file: C:\AWS-SCT-Demo\connection_example.csv [Browse]

[Download a connections file example](#)

Create AWS SCT projects for each source database

Add mapping rules to these projects and save conversion statistics for offline use

[Run] [Cancel]

2. Pilih Unduh contoh file koneksi untuk mengunduh templat kosong CSV file dengan parameter koneksi database.
3. Masukkan nilai untuk nama Proyek, Lokasi (untuk menyimpan laporan), dan file Koneksi (CSVfile).
4. Pilih Buat AWS SCT proyek untuk setiap database sumber untuk membuat proyek migrasi secara otomatis setelah membuat laporan penilaian.
5. Dengan mengaktifkan Buat AWS SCT proyek untuk setiap basis data sumber, Anda dapat memilih Tambahkan aturan pemetaan ke proyek ini dan menyimpan statistik konversi untuk penggunaan offline. Dalam hal ini, AWS SCT akan menambahkan aturan pemetaan untuk setiap proyek dan menyimpan metadata database sumber dalam proyek. Untuk informasi selengkapnya, lihat [Menggunakan mode offline di AWS Schema Conversion Tool](#).
6. Pilih Jalankan.

Bilah kemajuan muncul yang menunjukkan kecepatan penilaian basis data. Jumlah mesin target dapat memengaruhi runtime penilaian.

7. Pilih Ya jika pesan berikut ditampilkan: Analisis lengkap dari semua server Database mungkin memakan waktu. Apakah Anda ingin melanjutkan?

Ketika laporan penilaian multiserver selesai, layar muncul yang menunjukkan demikian.

8. Pilih Buka Laporan untuk melihat laporan penilaian ringkasan gabungan.

Secara default, AWS SCT menghasilkan laporan agregat untuk semua database sumber dan laporan penilaian terperinci untuk setiap nama skema dalam database sumber. Untuk informasi selengkapnya, lihat [Menemukan dan melihat laporan](#).

Dengan opsi Buat AWS SCT proyek untuk setiap database sumber diaktifkan, AWS SCT buat proyek kosong untuk setiap database sumber. AWS SCT juga membuat laporan penilaian seperti yang dijelaskan sebelumnya. Setelah Anda menganalisis laporan penilaian ini dan memilih tujuan migrasi untuk setiap database sumber, tambahkan database target ke proyek kosong ini.

Dengan opsi Tambahkan aturan pemetaan ke proyek ini dan simpan statistik konversi untuk penggunaan offline diaktifkan, AWS SCT buat proyek untuk setiap basis data sumber. Proyek-proyek ini mencakup informasi berikut:

- Database sumber Anda dan platform basis data target virtual. Untuk informasi selengkapnya, lihat [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).
- Aturan pemetaan untuk pasangan sumber-target ini. Untuk informasi selengkapnya, lihat [Pemetaan tipe data](#).
- Laporan penilaian migrasi database untuk pasangan sumber-target ini.
- Metadata skema sumber, yang memungkinkan Anda menggunakan AWS SCT proyek ini dalam mode offline. Untuk informasi selengkapnya, lihat [Menggunakan mode offline di AWS Schema Conversion Tool](#).

Mempersiapkan CSV file input

Untuk menyediakan parameter koneksi sebagai masukan untuk laporan penilaian multiserver, gunakan CSV file seperti yang ditunjukkan pada contoh berikut.

```
Name,Description,Secret Manager Key,Server IP,Port,Service Name,Database name,BigQuery
path,Source Engine,Schema Names,Use Windows Authentication,Login,Password,Use
SSL,Trust store,Key store,SSL authentication,Target Engines
Sales,,192.0.2.0,1521,pdb,,ORACLE,Q4_2021;FY_2021,,user,password,,,,,POSTGRESQL;AURORA_POSTGR
Marketing,,ec2-a-b-c-d.eu-
west-1.compute.amazonaws.com,1433,,target_audience,,MSSQL,customers.dbo,,user,password,,,,,AURORA
HR,,192.0.2.0,1433,,employees,,MSSQL,employees.%,true,,,,,AURORA_POSTGRESQL
Customers,,secret-name,,,,,MYSQL,customers,,,,,AURORA_POSTGRESQL
Analytics,,198.51.100.0,8195,,STATISTICS,,DB2LUW,BI_REPORTS,,user,password,,,,,POSTGRESQL
Products,,203.0.113.0,8194,,,,,TERADATA,new_products,,user,password,,,,,REDSHIFT
```

Contoh sebelumnya menggunakan titik koma untuk memisahkan dua nama skema untuk database. Sales Ini juga menggunakan titik koma untuk memisahkan dua platform migrasi database target untuk database. Sales

Juga, contoh sebelumnya digunakan AWS Secrets Manager untuk terhubung ke Customers database dan Windows Authentication untuk terhubung ke database. HR


Anda dapat membuat CSV file baru atau mengunduh templat untuk CSV file dari AWS SCT dan mengisi informasi yang diperlukan. Pastikan bahwa baris pertama CSV file Anda menyertakan nama kolom yang sama seperti yang ditunjukkan pada contoh sebelumnya.

Untuk men-download template dari CSV file input

1. Mulai AWS SCT.
2. Pilih File, lalu pilih Penilaian multiserver baru.
3. Pilih Unduh contoh file koneksi.

Pastikan CSV file Anda menyertakan nilai-nilai berikut, yang disediakan oleh template:

- Nama — Label teks yang membantu mengidentifikasi database Anda. AWS SCT menampilkan label teks ini dalam laporan penilaian.
- Deskripsi — Nilai opsional, di mana Anda dapat memberikan informasi tambahan tentang database.
- Secret Manager Key — Nama rahasia yang menyimpan kredensi database Anda di AWS Secrets Manager Untuk menggunakan Secrets Manager, pastikan Anda menyimpan AWS profil AWS SCT. Untuk informasi selengkapnya, lihat [Mengkonfigurasi AWS Secrets Manager di AWS Schema Conversion Tool](#).

 Important

AWS SCT mengabaikan parameter Kunci Manajer Rahasia jika Anda menyertakan parameter Server IP, Port, Login, dan Kata Sandi dalam file input.

- Server IP — Nama Layanan Nama Domain (DNS) atau alamat IP server basis data sumber Anda.
- Port — Port yang digunakan untuk terhubung ke server database sumber Anda.
- Nama Layanan — Jika Anda menggunakan nama layanan untuk terhubung ke database Oracle Anda, nama layanan Oracle untuk terhubung ke.

- Nama database — Nama database. Untuk database Oracle, gunakan Oracle System ID (). SID
- BigQuery path — path ke file kunci akun layanan untuk BigQuery database sumber Anda. Untuk informasi selengkapnya tentang membuat file ini, lihat [Hak istimewa untuk BigQuery sebagai sumber](#).
- Source Engine — Jenis database sumber Anda. Gunakan salah satu nilai berikut:
 - AZURE_ MSSQL untuk SQL Database Azure.
 - AZURE_ SYNAPSE untuk database Azure Synapse Analytics.
 - GOOGLE_ BIGQUERY untuk BigQuery database.
 - DB2ZOS untuk IBM Db2 untuk database z/OS.
 - DB2LUW untuk database IBM Db2LUW.
 - GREENPLUM untuk database Greenplum.
 - MSSQL untuk database Microsoft SQL Server.
 - MYSQL untuk SQL database saya.
 - NETEZZA untuk database Netezza.
 - ORACLE untuk database Oracle.
 - POSTGRES SQL untuk database PostgreSQL.
 - REDSHIFT untuk database Amazon Redshift.
 - SNOWFLAKE untuk database Snowflake.
 - SYBASE_ ASE untuk SAP ASE database.
 - TERADATA untuk database Teradata.
 - VERTICA untuk database Vertica.
- Nama Skema — Nama-nama skema database untuk dimasukkan dalam laporan penilaian.

Untuk Azure SQL Database, Azure Synapse Analytics BigQuery, Netezza,, Snowflake SAPASE, dan SQL Server, gunakan format nama skema berikut:

db_name.schema_name

Ganti *db_name* dengan nama database sumber.

Ganti *schema_name* dengan nama skema sumber.

Lampirkan nama database atau skema yang menyertakan titik dalam tanda kutip ganda seperti
Laporan penilaian multiserver yang ditunjukkan berikut: "database.name". "schema.name"

Pisahkan beberapa nama skema dengan menggunakan titik koma seperti yang ditunjukkan berikut: `Schema1;Schema2`

Nama database dan skema peka huruf besar/kecil.

Gunakan percent (%) sebagai wildcard untuk mengganti sejumlah simbol apa pun dalam database atau nama skema. Contoh sebelumnya menggunakan percent (%) sebagai wildcard untuk memasukkan semua skema dari `employees` database dalam laporan penilaian.

- Gunakan Otentikasi Windows - Jika Anda menggunakan Otentikasi Windows untuk menyambung ke database Microsoft SQL Server Anda, masukkan `true`. Untuk informasi selengkapnya, lihat [Menggunakan Otentikasi Windows saat menggunakan Microsoft SQL Server sebagai sumber](#).
- Login — Nama pengguna untuk terhubung ke server database sumber Anda.
- Kata sandi — Kata sandi untuk terhubung ke server basis data sumber Anda.
- Gunakan SSL - Jika Anda menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database sumber Anda, masukkan `true`.
- Toko kepercayaan — Toko kepercayaan yang digunakan untuk SSL koneksi Anda.
- Key store — Toko kunci yang digunakan untuk SSL koneksi Anda.
- SSLotentikasi — Jika Anda menggunakan SSL otentikasi berdasarkan sertifikat, masukkan `true`.
- Mesin Target — Platform basis data target. Gunakan nilai berikut untuk menentukan satu atau beberapa target dalam laporan penilaian:
 - `AURORA_MYSQL` untuk database Aurora My SQL -Compatible.
 - `AURORA_POSTGRESQL` untuk database Aurora SQL Postgre -Kompatibel.
 - `BABELFISH` untuk database Babelfish untuk Aurora Postgre. SQL
 - `MARIA_DB` untuk database MariaDB.
 - `MSSQL` untuk database Microsoft SQL Server.
 - `MYSQL` untuk SQL database saya.
 - `ORACLE` untuk database Oracle.
 - `POSTGRESQL` untuk database PostgreSQL.
 - `REDSHIFT` untuk database Amazon Redshift.

Pisahkan beberapa target dengan menggunakan titik koma seperti ini: `MYSQL;MARIA_DB` Jumlah target mempengaruhi waktu yang dibutuhkan untuk menjalankan penilaian.

Menemukan dan melihat laporan

Penilaian multiserver menghasilkan dua jenis laporan:

- Laporan agregat dari semua database sumber.
- Laporan penilaian rinci dari database target untuk setiap nama skema dalam database sumber.

Laporan disimpan di direktori yang Anda pilih untuk Lokasi di kotak dialog Penilaian multiserver baru.

Untuk mengakses laporan terperinci, Anda dapat menavigasi subdirektori, yang diatur oleh database sumber, nama skema, dan mesin basis data target.

Laporan agregat menampilkan informasi dalam empat kolom tentang kompleksitas konversi database target. Kolom mencakup informasi tentang konversi objek kode, objek penyimpanan, elemen sintaks, dan kompleksitas konversi.

Contoh berikut menunjukkan informasi untuk konversi dari dua skema database Oracle ke Amazon RDS untuk Postgre. SQL

Server IP address and port	Secret Manager key	Name	Description	Database name	Schema name	Code object conversion % for "Amazon RDS for PostgreSQL"	Storage object conversion % for "Amazon RDS for PostgreSQL"	Syntax Elements conversion % for "Amazon RDS for PostgreSQL"	Conversion Complexity for "Amazon RDS for PostgreSQL"
192.0.2.0:1521		Sales		ORCL	Q4_2021	97.78%	100.00%	98.76%	1
192.0.2.0:1521		Sales		pdb	FY_2021	82.35%	85.19%	99.24%	10

Empat kolom yang sama ditambahkan ke laporan untuk setiap mesin database target tambahan yang ditentukan.

Untuk detail tentang cara membaca informasi ini, lihat berikut.

Output untuk laporan penilaian agregat

Laporan penilaian migrasi database multiserver agregat di AWS Schema Conversion Tool adalah CSV file dengan kolom berikut:

- Server IP address and port
- Secret Manager key
- Name
- Description
- Database name

- Schema name
- Code object conversion % for *target_database*
- Storage object conversion % for *target_database*
- Syntax elements conversion % for *target_database*
- Conversion complexity for *target_database*

Untuk mengumpulkan informasi, AWS SCT jalankan laporan penilaian lengkap dan kemudian agregat laporan berdasarkan skema.

Dalam laporan tersebut, tiga bidang berikut menunjukkan persentase kemungkinan konversi otomatis berdasarkan penilaian:

Konversi objek kode%

Persentase objek kode dalam skema yang AWS SCT dapat dikonversi secara otomatis atau dengan perubahan minimal. Objek kode termasuk prosedur, fungsi, tampilan, dan sejenisnya.

Konversi objek penyimpanan%

Persentase objek penyimpanan yang SCT dapat dikonversi secara otomatis atau dengan sedikit perubahan. Objek penyimpanan termasuk tabel, indeks, kendala, dan sejenisnya.

Elemen sintaks konversi%

Persentase elemen sintaks yang SCT dapat dikonversi secara otomatis. Elemen sintaks meliputi `SELECT`, `FROM`, `DELETE`, dan `JOIN` klausa, dan serupa.

Perhitungan kompleksitas konversi didasarkan pada gagasan item tindakan. Item tindakan mencerminkan jenis masalah yang ditemukan dalam kode sumber yang perlu Anda perbaiki secara manual selama migrasi ke target tertentu. Item tindakan dapat memiliki beberapa kejadian.

Timbangan tertimbang mengidentifikasi tingkat kompleksitas untuk melakukan migrasi. Angka 1 mewakili tingkat kompleksitas terendah, dan angka 10 mewakili tingkat kompleksitas tertinggi.

Mengonversi skema database di AWS Schema Conversion Tool

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skema database yang ada dari satu mesin database ke mesin database lainnya. Mengonversi database menggunakan antarmuka AWS SCT pengguna bisa cukup sederhana, tetapi ada beberapa hal yang perlu dipertimbangkan sebelum Anda melakukan konversi.

Misalnya, Anda dapat menggunakan AWS SCT untuk melakukan hal berikut:

- Anda dapat menggunakan AWS SCT untuk menyalin skema database lokal yang ada ke instans Amazon RDS DB yang menjalankan mesin yang sama. Anda dapat menggunakan fitur ini untuk menganalisis potensi penghematan biaya pindah ke cloud dan mengubah jenis lisensi Anda.
- Dalam beberapa kasus, fitur database tidak dapat dikonversi ke RDS fitur Amazon yang setara. Jika Anda meng-host dan mengelola sendiri database di platform Amazon Elastic Compute Cloud EC2 (Amazon), Anda dapat meniru fitur-fitur ini dengan mengganti AWS layanan untuk mereka.
- AWS SCT mengotomatiskan sebagian besar proses konversi skema database pemrosesan transaksi online Anda (OLTP) ke instans DB SQL Saya Amazon Relational Database Service (Amazon), kluster DB RDS Amazon Aurora, atau instans Postgre DB. SQL Mesin basis data sumber dan target berisi banyak fitur dan kemampuan yang berbeda, dan AWS SCT berupaya membuat skema yang setara di instans Amazon RDS DB Anda sedapat mungkin. Jika tidak ada konversi langsung yang memungkinkan, AWS SCT berikan daftar tindakan yang mungkin untuk Anda ambil.

Topik

- [Menerapkan aturan migrasi di AWS Schema Conversion Tool](#)
- [Mengkonversi skema usinf AWS SCT](#)
- [Mengonversi skema secara manual AWS SCT](#)
- [Memperbarui dan menyegarkan skema yang dikonversi di AWS SCT](#)
- [Menyimpan dan menerapkan skema yang dikonversi di AWS SCT](#)
- [Membandingkan skema di AWS Schema Conversion Tool](#)
- [Melihat objek yang ditransformasikan terkait di AWS Schema Conversion Tool](#)

AWS SCT mendukung konversi pemrosesan transaksi online (OLTP) berikut.

Basis data sumber	Basis data target
IBMDB2 untuk z/OS (versi 12)	Amazon Aurora My SQL -Compatible Edition, Amazon Aurora Postgre -Edisi Kompatibel, My, Postgre SQL SQL SQL
IBMDB2 LUW (versi 9.1, 9.5, 9.7, 10.5, 11.1, dan 11.5)	Aurora Saya, Aurora Postgre, SQL MariaDB, Saya, SQL Postgre SQL SQL
Database Microsoft Azure SQL	Aurora Saya, Aurora SQL Postgre, Saya, Postgre SQL SQL SQL
Microsoft SQL Server (versi 2008 R2 dan lebih tinggi)	Aurora MySQL, Aurora PostgreSQL, Babelfish untuk Aurora Postgre, MariaDB, Microsoft Server, My, SQL Postgre SQL SQL SQL
Saya SQL (versi 5.5 dan lebih tinggi)	Aurora Postgre, Saya, Postgre SQL SQL SQL Anda dapat memigrasikan skema dan data dari My SQL ke cluster Aurora My SQL DB tanpa menggunakan AWS SCT. Untuk informasi selengkapnya, lihat Memigrasi data ke kluster DB Amazon Aurora .
Oracle (versi 10.2 dan lebih tinggi)	Aurora Saya, Aurora Postgre, SQL MariaDB, Saya, SQL Oracle, Postgre SQL SQL
Postgre SQL (versi 9.1 dan lebih tinggi)	Aurora Saya, Aurora SQL Postgre, Saya, Postgre SQL SQL SQL
SAPASE(12,5, 15,0, 15,5, 15,7, dan 16,0)	Aurora Saya, Aurora Postgre, SQL MariaDB, Saya, SQL Postgre SQL SQL

Untuk informasi tentang mengonversi skema gudang data, lihat [Mengonversi skema gudang data ke Amazon menggunakan RDS AWS SCT](#)

Untuk mengonversi skema database Anda ke AmazonRDS, Anda mengambil langkah-langkah tingkat tinggi berikut:

- [Membuat aturan migrasi di AWS SCT](#)— Sebelum Anda mengonversi skema AWS SCT, Anda dapat mengatur aturan yang mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek.
- [???](#)— AWS SCT membuat versi lokal dari skema yang dikonversi untuk Anda tinjau, tetapi itu tidak menerapkannya ke instans DB target Anda sampai Anda siap.
- [Menggunakan laporan penilaian di AWS Schema Conversion Tool](#)— AWS SCT membuat laporan penilaian migrasi database yang merinci elemen skema yang tidak dapat dikonversi secara otomatis. Anda dapat menggunakan laporan ini untuk mengidentifikasi di mana Anda perlu membuat skema di instans Amazon RDS DB yang kompatibel dengan basis data sumber Anda.
- [Mengkonversi skema usinf AWS SCT](#)— Jika Anda memiliki elemen skema yang tidak dapat dikonversi secara otomatis, Anda memiliki dua pilihan: perbarui skema sumber dan kemudian konversi lagi, atau buat elemen skema yang setara di instans DB Amazon RDS target Anda.
- [Memperbarui dan menyegarkan skema yang dikonversi di AWS SCT](#)— Anda dapat memperbarui AWS SCT proyek Anda dengan skema terbaru dari database sumber Anda.
- [Menyimpan dan menerapkan skema yang dikonversi di AWS SCT](#)— Saat Anda siap, AWS SCT terapkan skema yang dikonversi dalam proyek lokal Anda ke instans Amazon RDS DB target Anda.

Menerapkan aturan migrasi di AWS Schema Conversion Tool

Sebelum mengonversi skema AWS SCT, Anda dapat mengatur aturan migrasi. Aturan migrasi AWS SCT dapat melakukan transformasi seperti mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek. Misalnya, anggaplah Anda memiliki satu set tabel dalam skema sumber Anda bernama `test_TABLE_NAME`. Anda dapat mengatur aturan yang mengubah awalan `test_` ke awalan `demo_` dalam skema target.

Note

Anda hanya dapat membuat aturan migrasi untuk mesin basis data sumber dan target yang berbeda.

Anda dapat membuat aturan migrasi yang melakukan tugas-tugas berikut:

- Menambahkan, menghapus, atau mengganti awalan
- Menambahkan, menghapus, atau mengganti sufiks
- Ubah pemeriksaan kolom
- Ubah tipe data
- Mengubah panjang `char`, `varchar`, `nvarchar`, dan tipe `string` data
- Pindahkan objek
- Ganti nama objek

Anda dapat membuat aturan migrasi untuk objek berikut:

- Basis Data
- Skema
- Tabel
- Kolom

Membuat aturan migrasi

Anda dapat membuat aturan migrasi dan menyimpan aturan sebagai bagian dari proyek Anda. Dengan proyek Anda terbuka, gunakan prosedur berikut untuk membuat aturan migrasi.

Untuk membuat aturan migrasi

1. Pada menu Tampilan, pilih Tampilan pemetaan.
2. Dalam pemetaan Server, pilih sepasang server sumber dan target.
3. Pilih Aturan migrasi baru. Kotak dialog aturan Transformasi muncul.
4. Pilih Tambahkan aturan baru. Baris baru ditambahkan ke daftar aturan.
5. Konfigurasi aturan Anda:
 - a. Untuk Nama, masukkan nama untuk topik Anda.
 - b. Untuk Untuk, pilih jenis objek yang aturan berlaku.
 - c. Untuk tempat, masukkan filter untuk diterapkan ke objek sebelum menerapkan aturan migrasi. Klausula `where` dievaluasi dengan menggunakan klausula `like`. Anda dapat memasukkan nama yang tepat untuk memilih satu objek, atau Anda dapat memasukkan pola untuk memilih beberapa objek.

Bidang yang tersedia untuk klausa where berbeda tergantung pada jenis objek. Misalnya, jika jenis objek adalah skema hanya ada satu bidang yang tersedia, untuk nama skema.

- d. Untuk Tindakan, pilih jenis aturan migrasi yang ingin Anda buat.
- e. Tergantung pada jenis aturan, masukkan satu atau dua nilai tambahan. Misalnya, untuk mengganti nama objek, masukkan nama baru objek. Untuk mengganti awalan, masukkan awalan lama dan awalan baru.

Untuk tipe data char, varchar, nvarchar, dan string, Anda dapat mengubah panjang tipe data menggunakan operator perkalian. Misalnya, `%*4` nilai mengubah tipe `varchar(10)` data menjadi `varchar(40)`.

6. Setelah mengonfigurasi aturan migrasi, pilih Simpan untuk menyimpan aturan Anda. Anda juga dapat memilih Batal untuk membatalkan perubahan Anda.

Transformation rules affect how the converted objects to be named on the target database. For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority. Default transformation rules are always at the top of the list and can be disabled or changed only in the [Conversion settings](#) tab. The rules can be exported to a file for later use in the DMS, but please note that AWS DMS [doesn't support](#) more than one transformation rule per schema level or per table level. Note, every rule might have to following status along with the corresponding color:

- Successfully created enabled rule
- Rule with incorrect data entered

Transformation rule: For **tables** where database name is like '%' and schema name is like '%' and table name is like 'test_%' add prefix 'demo_%'

Name: Transformation rule

For: table

where database name like: % schema name like: % table name like: test_%

Actions: add prefix demo_%

Buttons: Save, Cancel, Add new rule, Export script for DMS, Import script into SCT, Save all, Close

7. Setelah Anda selesai menambahkan, mengedit, dan menghapus aturan, pilih Simpan Semua untuk menyimpan semua perubahan Anda.
8. Pilih Tutup untuk menutup kotak dialog Aturan transformasi.

Anda dapat menggunakan ikon sakelar untuk menonaktifkan aturan migrasi tanpa menghapusnya. Anda dapat menggunakan ikon salin untuk menduplikasi aturan migrasi yang ada. Anda dapat menggunakan ikon pensil untuk mengedit aturan migrasi yang ada. Anda dapat menggunakan ikon

hapus untuk menghapus aturan migrasi yang ada. Untuk menyimpan perubahan apa pun yang Anda buat pada aturan migrasi, pilih Simpan Semua.

Mengekspor aturan migrasi

Jika digunakan AWS DMS untuk memigrasikan data dari database sumber ke database target, Anda dapat memberikan informasi tentang aturan migrasi ke AWS DMS. Untuk informasi selengkapnya tentang tugas, lihat [Bekerja dengan tugas AWS Database Migration Service replikasi](#).

Untuk mengekspor aturan migrasi

1. Di dalam AWS Schema Conversion Tool, pilih Mapping View pada menu View.
2. Di Aturan migrasi, pilih aturan migrasi, lalu pilih Ubah aturan migrasi.
3. Pilih Ekspor skrip untuk AWS DMS.
4. Jelajahi lokasi tempat Anda ingin menyimpan skrip, lalu pilih Simpan. Aturan migrasi Anda disimpan sebagai JSON skrip yang dapat digunakan oleh AWS DMS.

Mengkonversi skema usinf AWS SCT

Setelah Anda menghubungkan proyek Anda ke database sumber dan instans Amazon RDS DB target Anda, AWS Schema Conversion Tool proyek Anda akan menampilkan skema dari database sumber Anda di panel kiri. Skema disajikan dalam format tampilan pohon, dan setiap simpul pohon dimuat dengan lambat. Ketika Anda memilih node dalam tampilan pohon, AWS SCT meminta informasi skema dari database sumber Anda pada saat itu.

Anda dapat memilih item skema dari database sumber Anda dan kemudian mengonversi skema menjadi skema setara untuk mesin DB dari instans DB target Anda. Anda dapat memilih item skema apa pun dari database sumber Anda untuk dikonversi. Jika item skema yang Anda pilih bergantung pada item induk, maka buat AWS SCT juga skema untuk item induk. Misalnya, anggaplah Anda memilih tabel untuk dikonversi. Jika demikian, AWS SCT menghasilkan skema untuk tabel, dan database tempat tabel berada.

Mengubah skema

Untuk mengonversi skema dari basis data sumber Anda, pilih kotak centang untuk nama skema yang akan dikonversi. Selanjutnya, pilih skema ini dari panel kiri proyek Anda. AWS SCT menyoroti nama skema dengan warna biru. Buka menu konteks (klik kanan) untuk skema, dan pilih Konversi skema, seperti yang ditunjukkan berikut.

File Actions Main view Settings Applications Help Add source Add target

Connected. Click to disconnect

Servers

- SQL Server - ec2-52-17-21-76.eu-west-1.compute.am
 - Databases [12]
 - AdventureWorks2012_CS
 - alfresco
 - GOLD_TEST_SS_PG
 - LARGE_DB_SS
 - master
 - model
 - msdb
 - tempdb
 - TEST**
 - vmap
 - vpas
 - vrecon
 - Server Objects
 - SQL Server Agent
 - Applications
 - SQL Scripts
 - noSQL Clusters
 - ETL

Create mapping...
 Create report
Convert schema
 Register agent
 Compare schema
 Load schema
 Hide schema
 Refresh from database
 Collect statistics
 Upload statistics
 Create DMS task
 Create Local & DMS task
 Create Local task
 Add virtual partitioning
 Save as SQL

Properties SQL Related converted objects Statistics

Name	
Created or last modified	
Created	2021-09-06 09:56:08.26
Object name	
Name	TEST
compatibility-level	100
collation-name	SQL_Latin1_General_CP1_CI_AS

Properties SQL Apply status Key management

Name	
Category	
Name	<Aurora_MySQL (virtual)>

Setelah Anda mengonversi skema dari database sumber Anda, Anda dapat memilih item skema dari panel kiri proyek Anda dan melihat skema yang dikonversi di panel tengah proyek Anda. Panel tengah bawah menampilkan properti dan SQL perintah untuk membuat skema yang dikonversi, seperti yang ditunjukkan berikut.

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the source database structure: SQL Server - ec2-52-17-21-76.eu-west-1.cc > Databases [12] > LARGE_DB_SS > Schemas [2] > dbo > Tables [4] > Account. The main pane shows the SQL definition for the [Account] table in the source database:

```

1 CREATE TABLE [dbo].[Account] (
2 [ID] numeric(14,0) NOT NULL,
3 [AccountNo] varchar(16) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
4 [CurrencyID] numeric(3,0) NOT NULL,
5 [Description] varchar(160) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
6 [CustomerID] numeric(14,0) NOT NULL,
7 [StateID] numeric(2,0) NOT NULL,
8 [AccountBalance] numeric(14,3) NOT NULL,
9 [BlockedAmount] numeric(14,3) NOT NULL,
10 [Opendate] datetime NULL,
11 [Closedate] datetime NULL,
12 [RespManagerID] numeric(5,0) NULL,
13 [BankID] varchar(10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
14 )
15 ON [PRIMARY];

```

Below the source SQL, the target Amazon RDS instance is identified as 'Target Amazon RDS for MySQL table: Account'. The converted SQL definition for the target instance is shown:

```

1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo VARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,

```

Setelah Anda mengubah skema Anda, Anda dapat menyimpan proyek Anda. Informasi skema dari database sumber Anda disimpan dengan proyek Anda. Fungsionalitas ini berarti Anda dapat bekerja offline tanpa terhubung ke database sumber Anda. AWS SCT menghubungkan ke database sumber Anda untuk memperbarui skema dalam proyek Anda jika Anda memilih Refresh from Database untuk database sumber Anda. Untuk informasi selengkapnya, lihat [Memperbarui dan menyegarkan skema yang dikonversi di AWS SCT](#).

Anda dapat membuat laporan penilaian migrasi database dari item yang tidak dapat dikonversi secara otomatis. Laporan penilaian berguna untuk mengidentifikasi dan menyelesaikan item skema yang tidak dapat dikonversi secara otomatis. Untuk informasi selengkapnya, lihat [Menggunakan laporan penilaian di AWS Schema Conversion Tool](#).

Saat AWS SCT menghasilkan skema yang dikonversi, itu tidak segera menerapkannya ke instans DB target. Sebagai gantinya, skema yang dikonversi disimpan secara lokal sampai Anda siap menerapkannya ke instans DB target. Untuk informasi selengkapnya, lihat [Menerapkan skema konversi Anda](#).

Mengedit skema yang dikonversi

Anda dapat mengedit skema yang dikonversi dan menyimpan perubahan sebagai bagian dari proyek Anda.

Untuk mengedit skema yang dikonversi

1. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih item skema yang ingin Anda edit skema yang dikonversi.
2. Di panel tengah bawah yang menampilkan skema yang dikonversi untuk item yang dipilih, pilih tab. SQL
3. Dalam teks yang ditampilkan untuk SQLtab, ubah skema sesuai kebutuhan. Skema secara otomatis disimpan dengan proyek Anda saat Anda memperbaruinya.

```
▼ Target Amazon RDS for MySQL table: Account
Properties SQL Columns Partition columns Apply status Key management
1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo NVARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,
13 BankID VARCHAR(10) NOT NULL
14 );
```

Perubahan yang Anda buat pada skema yang dikonversi disimpan dengan proyek Anda saat Anda melakukan pembaruan. Jika Anda baru mengonversi item skema dari basis data sumber Anda, dan Anda telah membuat pembaruan ke skema yang dikonversi sebelumnya untuk item tersebut, pembaruan yang ada akan digantikan oleh item skema yang baru dikonversi berdasarkan basis data sumber Anda.

Membersihkan skema yang dikonversi

Sampai Anda menerapkan skema ke instans DB target Anda, AWS SCT hanya menyimpan skema yang dikonversi secara lokal di proyek Anda. Anda dapat menghapus skema yang direncanakan dari proyek Anda dengan memilih node tampilan pohon untuk instans DB Anda, lalu memilih Refresh dari Database. Karena tidak ada skema yang ditulis ke instans DB target Anda, penyegaran dari database akan menghapus elemen skema yang direncanakan dalam AWS SCT proyek Anda agar sesuai dengan apa yang ada di instans DB sumber Anda.

Mengonversi skema secara manual AWS SCT

Laporan penilaian menyertakan daftar item yang tidak dapat dikonversi secara otomatis ke mesin database instans Amazon RDS DB target Anda. Untuk setiap item yang tidak dapat dikonversi, ada item tindakan di Item Tindakan tab.

Anda dapat menanggapi item tindakan dalam laporan penilaian dengan cara berikut:

- Ubah skema basis data sumber Anda.
- Ubah skema basis data target Anda.

Memodifikasi skema sumber Anda

Untuk beberapa item, mungkin lebih mudah untuk memodifikasi skema database dalam database sumber Anda ke skema yang dapat dikonversi secara otomatis. Pertama, verifikasi bahwa perubahan baru kompatibel dengan arsitektur aplikasi Anda, lalu perbarui skema di database sumber Anda. Terakhir, segarkan proyek Anda dengan informasi skema yang diperbarui. Anda kemudian dapat mengonversi skema yang diperbarui, dan membuat laporan penilaian migrasi database baru. Item tindakan tidak lagi muncul untuk item yang berubah dalam skema sumber.

Keuntungan dari proses ini adalah skema Anda yang diperbarui selalu tersedia saat Anda menyegarkan dari basis data sumber Anda.

Memodifikasi skema target Anda

Untuk beberapa item, mungkin lebih mudah untuk menerapkan skema yang dikonversi ke database target Anda, dan kemudian menambahkan item skema yang setara secara manual ke database target Anda untuk item yang tidak dapat dikonversi secara otomatis. Anda dapat menulis semua skema yang dapat dikonversi secara otomatis ke instans DB target Anda dengan menerapkan

skema. Untuk informasi selengkapnya, lihat [Menyimpan dan menerapkan skema yang dikonversi di AWS SCT](#).

Skema yang ditulis ke instans DB target Anda tidak berisi item yang tidak dapat dikonversi secara otomatis. Setelah menerapkan skema ke instans DB target Anda, Anda kemudian dapat secara manual membuat skema dalam instance DB target Anda yang setara dengan yang ada di database sumber. Item tindakan dalam laporan penilaian migrasi database berisi saran tentang cara membuat skema yang setara.

Warning

Jika Anda membuat skema secara manual di instans DB target Anda, simpan salinan pekerjaan manual apa pun yang Anda lakukan. Jika Anda menerapkan skema yang dikonversi dari proyek Anda ke instans DB target Anda lagi, itu menimpa pekerjaan manual yang telah Anda lakukan.

Dalam beberapa kasus, Anda tidak dapat membuat skema setara dalam instans DB target Anda. Anda mungkin perlu merancang ulang sebagian aplikasi dan database Anda untuk menggunakan fungsionalitas yang tersedia dari mesin DB untuk instans DB target Anda. Dalam kasus lain, Anda dapat mengabaikan skema yang tidak dapat dikonversi secara otomatis.

Memperbarui dan menyegarkan skema yang dikonversi di AWS SCT

Anda dapat memperbarui skema sumber dan skema target dalam proyek Anda AWS Schema Conversion Tool .

- **Sumber** - Jika Anda memperbarui skema untuk database sumber Anda, AWS SCT ganti skema dalam proyek Anda dengan skema terbaru dari database sumber Anda. Dengan menggunakan fungsi ini, Anda dapat memperbarui proyek Anda jika perubahan telah dilakukan pada skema database sumber Anda.
- **Target** - Jika Anda memperbarui skema untuk instans RDS DB Amazon target Anda, AWS SCT ganti skema dalam proyek Anda dengan skema terbaru dari instans DB target Anda. Jika Anda belum menerapkan skema apa pun ke instans DB target Anda, AWS SCT hapus skema yang dikonversi dari proyek Anda. Anda kemudian dapat mengonversi skema dari database sumber Anda untuk instance DB target yang bersih.

Anda memperbarui skema dalam AWS SCT proyek Anda dengan memilih Refresh dari Database.

Note

Saat Anda menyegarkan skema, AWS SCT memuat metadata hanya sesuai kebutuhan. Untuk sepenuhnya memuat semua skema database Anda, buka menu konteks (klik kanan) untuk skema Anda, dan pilih Muat skema. Misalnya, Anda dapat menggunakan opsi ini untuk memuat metadata untuk database Anda sekaligus, dan kemudian bekerja secara offline.

Menyimpan dan menerapkan skema yang dikonversi di AWS SCT

Saat AWS Schema Conversion Tool menghasilkan skema yang dikonversi (seperti yang ditunjukkan pada [????](#)), skema tersebut tidak segera menerapkan skema yang dikonversi ke instans DB target. Sebagai gantinya, skema yang dikonversi disimpan secara lokal di proyek Anda sampai Anda siap menerapkannya ke instans DB target. Dengan menggunakan fungsi ini, Anda dapat bekerja dengan item skema yang tidak dapat dikonversi secara otomatis ke mesin DB target Anda. Untuk informasi selengkapnya tentang item yang tidak dapat dikonversi secara otomatis, lihat [Menggunakan laporan penilaian di AWS Schema Conversion Tool](#).

Anda dapat secara opsional meminta alat menyimpan skema yang dikonversi ke file sebagai SQL skrip sebelum menerapkan skema ke instans DB target Anda. Anda juga dapat meminta alat menerapkan skema yang dikonversi langsung ke instans DB target Anda.

Menyimpan skema yang telah dikonversi ke file

Anda dapat menyimpan skema yang dikonversi sebagai SQL skrip dalam file teks. Dengan menggunakan pendekatan ini, Anda dapat memodifikasi SQL skrip yang dihasilkan dari AWS SCT ke item alamat yang alat tidak dapat dikonversi secara otomatis. Anda kemudian dapat menjalankan skrip yang diperbarui pada instans DB target Anda untuk menerapkan skema yang dikonversi ke basis data target Anda.

Untuk menyimpan skema yang dikonversi sebagai skrip SQL

1. Pilih skema Anda dan buka menu konteks (klik kanan).
2. Pilih Simpan sebagai SQL.
3. Masukkan nama file dan pilih Simpan.
4. Simpan skema Anda yang telah dikonversi menggunakan salah satu opsi berikut:

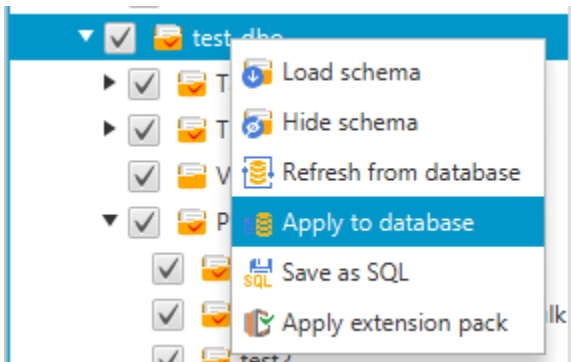
- Berkas tunggal
- File tunggal per tahap
- File tunggal per pernyataan

Untuk memilih format SQL skrip

1. Pada menu Pengaturan, pilih Pengaturan proyek.
2. Pilih Simpan skrip.
3. Untuk Vendor, pilih platform database.
4. Untuk Menyimpan SQL skrip, pilih bagaimana Anda ingin menyimpan skrip skema database Anda.
5. Pilih OK untuk menyimpan pengaturan.

Menerapkan skema konversi Anda

Saat Anda siap menerapkan skema yang dikonversi ke instans RDS DB Amazon target, pilih elemen skema dari panel kanan proyek Anda. Buka menu konteks (klik kanan) untuk elemen skema, lalu pilih Terapkan ke database, seperti yang ditunjukkan berikut.



Skema paket ekstensi

Pertama kali Anda menerapkan skema yang dikonversi ke instans DB target Anda, AWS SCT tambahkan skema tambahan ke instans DB target Anda. Skema ini mengimplementasikan fungsi sistem dari database sumber yang diperlukan saat menulis skema yang dikonversi ke instans DB target Anda. Skema ini disebut skema paket ekstensi.

Jangan mengubah skema paket ekstensi, atau Anda mungkin menemukan hasil yang tidak terduga dalam skema konversi yang ditulis ke instans DB target Anda. Ketika skema Anda sepenuhnya

dimigrasikan ke instans DB target Anda, dan Anda tidak perlu lagi AWS SCT, Anda dapat menghapus skema paket ekstensi.

Skema paket ekstensi diberi nama sesuai dengan basis data sumber Anda sebagai berikut:

- IBM Db2 LUW: `aws_db2_ext`
- SQL Server Microsoft: `aws_sqlserver_ext`
- SayaSQL: `aws_mysql_ext`
- Oracle: `aws_oracle_ext`
- PostgreSQL: `aws_postgresql_ext`
- SAP ASE: `aws_sapase_ext`

Untuk informasi selengkapnya, lihat [Menggunakan AWS Lambda fungsi dari paket AWS SCT ekstensi](#).

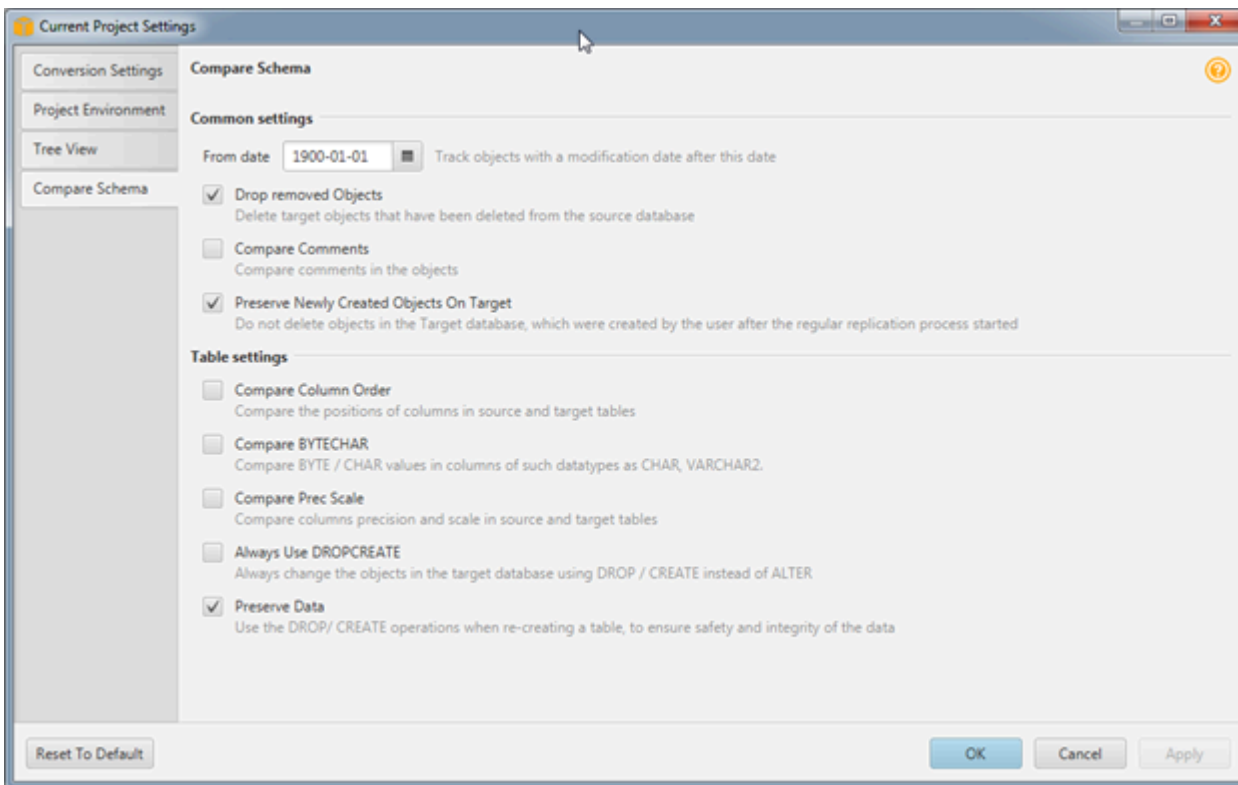
Membandingkan skema di AWS Schema Conversion Tool

Jika Anda membuat perubahan pada skema sumber atau target setelah bermigrasi, Anda dapat membandingkan dua skema database menggunakan AWS SCT. Anda dapat membandingkan skema untuk versi yang sama dengan atau lebih awal dari skema sumber.

Perbandingan skema berikut didukung:

- Oracle ke Oracle, versi 12.1.0.2.0, 11.1.0.7.0, 11.2.0.1.0, 10
- SQL Server ke SQL Server, versi 2016, 2014, 2012, 2008RD2, 2008
- PostgreSQL ke PostgreSQL dan Aurora PostgreSQL -Edisi yang Kompatibel, versi 9.6, 9.5.9, 9.5.4
- MySQL ke MySQL, versi 5.6.36, 5.7.17, 5.5

Anda menentukan pengaturan untuk perbandingan skema pada tab Bandingkan Skema pada halaman Pengaturan Proyek.

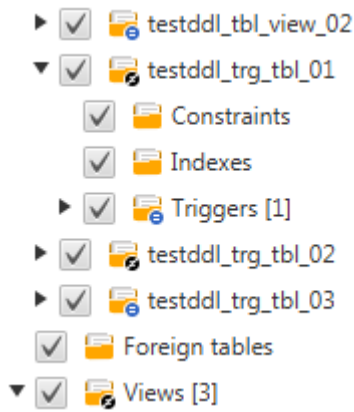


Untuk membandingkan skema, Anda memilih skema, dan AWS SCT menunjukkan objek yang berbeda antara dua skema dan objek yang tidak.

Untuk membandingkan dua skema

1. Buka AWS SCT proyek yang sudah ada, atau buat proyek dan sambungkan ke titik akhir sumber dan target.
2. Pilih skema yang ingin Anda bandingkan.
3. Buka menu konteks (klik kanan) dan pilih Bandingkan Skema.

AWS SCT menunjukkan objek yang berbeda antara dua skema dengan menambahkan lingkaran hitam ke ikon objek.



Anda dapat menerapkan hasil perbandingan skema ke satu objek, ke satu kategori objek, atau ke seluruh skema. Pilih kotak di samping kategori, objek, atau skema yang ingin Anda terapkan hasilnya.

Melihat objek yang ditransformasikan terkait di AWS Schema Conversion Tool

Setelah konversi skema, dalam beberapa kasus AWS SCT mungkin telah menciptakan beberapa objek untuk satu objek skema pada database sumber. Misalnya, ketika melakukan SQL konversi Oracle ke Postgre, AWS SCT mengambil setiap pemicu Oracle dan mengubahnya menjadi fungsi pemicu dan pemicu pada target Postgre. SQL Juga, ketika AWS SCT mengkonversi fungsi paket Oracle atau prosedur ke PostgreSQL, itu menciptakan fungsi yang setara dan INIT fungsi yang harus dijalankan sebagai blok init sebelum prosedur atau fungsi dapat dijalankan.

Prosedur berikut memungkinkan Anda melihat semua objek terkait yang dibuat setelah konversi skema.

Untuk melihat objek terkait yang dibuat selama konversi skema

1. Setelah konversi skema, pilih objek yang dikonversi dalam tampilan pohon target.
2. Pilih tab Objek Terkonversi Terkait.
3. Lihat daftar objek target terkait.

Mengonversi skema gudang data ke Amazon menggunakan RDS AWS SCT

AWS Schema Conversion Tool (AWS SCT) mengotomatiskan sebagian besar proses konversi skema gudang data Anda ke skema database Amazon atau RDS Aurora. Karena mesin basis data sumber dan target dapat memiliki banyak fitur dan kemampuan yang berbeda, cobalah AWS SCT untuk membuat skema yang setara dalam basis data target Anda sedapat mungkin. Jika tidak ada konversi langsung yang memungkinkan, AWS SCT berikan laporan penilaian dengan daftar tindakan yang mungkin untuk Anda ambil. Menggunakan AWS SCT, Anda dapat mengelola kunci, memetakan tipe data dan objek, dan membuat konversi manual.

AWS SCT dapat mengonversi skema gudang data berikut.

- Amazon Redshift
- Azure Synapse Analytics (versi 10)
- BigQuery
- Database Greenplum (versi 4.3)
- Microsoft SQL Server (versi 2008 dan lebih tinggi)
- Netezza (versi 7.0.3 dan lebih tinggi)
- Oracle (versi 10.2 dan lebih tinggi)
- Kepingan salju (versi 3)
- Teradata (versi 13 dan lebih tinggi)
- Vertica (versi 7.2 dan lebih tinggi)

Untuk informasi tentang mengonversi skema basis data pemrosesan transaksi online (OLTP), lihat.

[Mengonversi skema database di AWS Schema Conversion Tool](#)

Untuk mengonversi skema gudang data, lakukan langkah-langkah berikut:

1. Tentukan strategi dan aturan pengoptimalan, dan tentukan aturan migrasi yang AWS SCT ingin Anda gunakan. Anda dapat mengatur aturan yang mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek.

Anda dapat menentukan aturan pengoptimalan dan migrasi di Pengaturan. Untuk informasi selengkapnya tentang strategi pengoptimalan, [Memilih strategi dan aturan pengoptimalan untuk](#)

- [digunakan AWS SCT](#) lihat. untuk informasi selengkapnya tentang aturan migrasi, lihat [Membuat aturan migrasi di AWS SCT](#)
2. Berikan statistik pada gudang data sumber Anda sehingga AWS SCT dapat mengoptimalkan bagaimana gudang data Anda dikonversi. Anda dapat mengumpulkan statistik langsung dari database, atau mengunggah file statistik yang ada. Untuk informasi selengkapnya tentang penyediaan statistik gudang data, lihat [Mengumpulkan atau mengunggah statistik untuk AWS SCT](#).
 3. Buat laporan penilaian migrasi database yang merinci elemen skema yang tidak dapat dikonversi secara otomatis. Anda dapat menggunakan laporan ini untuk mengidentifikasi di mana Anda perlu membuat skema secara manual di database target Anda yang kompatibel dengan basis data sumber Anda. Untuk informasi lebih lanjut tentang laporan penilaian, lihat [Menggunakan laporan penilaian di AWS Schema Conversion Tool](#).
 4. Konversi skema. AWS SCT membuat versi lokal dari skema yang dikonversi untuk Anda tinjau, tetapi itu tidak menerapkannya ke database target Anda sampai Anda siap. Untuk informasi selengkapnya tentang mengonversi, lihat [Mengonversi skema Anda menggunakan AWS SCT](#)
 5. Setelah mengonversi skema, Anda dapat mengelola dan mengedit kunci Anda. Manajemen kunci adalah jantung dari konversi gudang data. Untuk informasi selengkapnya tentang mengelola kunci, lihat [Mengelola dan menyesuaikan kunci di AWS SCT](#).
 6. Jika Anda memiliki elemen skema yang tidak dapat dikonversi secara otomatis, Anda memiliki dua pilihan: memperbarui skema sumber dan kemudian mengonversi lagi, atau membuat elemen skema yang setara dalam database target Anda. Untuk informasi selengkapnya tentang mengonversi elemen skema secara manual, lihat. [Menangani konversi manual di AWS SCT](#) Untuk informasi selengkapnya tentang memperbarui skema sumber Anda, lihat [Memperbarui dan menyegarkan skema konversi Anda di AWS SCT](#).
 7. Ketika Anda siap, Anda dapat menerapkan skema yang dikonversi ke database target Anda. Untuk informasi selengkapnya tentang menyimpan dan menerapkan skema yang dikonversi, lihat [Menyimpan dan menerapkan skema konversi Anda di AWS SCT](#).

Izin untuk Amazon Redshift sebagai target

Izin yang diperlukan untuk Amazon Redshift sebagai target tercantum sebagai berikut:

- CREATEON DATABASE — memungkinkan untuk membuat skema baru dalam database.
- CREATEON SCHEMA — memungkinkan untuk membuat objek dalam skema database.
- GRANTUSAGEON LANGUAGE — memungkinkan untuk membuat fungsi dan prosedur baru dalam database.

- GRANTSELECTON ALL TABLES IN SCHEMA pg_catalog — memberi pengguna informasi sistem tentang cluster Amazon Redshift.
- GRANTSELECTON pg_class_info - menyediakan pengguna dengan informasi tentang gaya distribusi tabel.

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

```
CREATE USER user_name PASSWORD your_password;  
GRANT CREATE ON DATABASE db_name TO user_name;  
GRANT CREATE ON SCHEMA schema_name TO user_name;  
GRANT USAGE ON LANGUAGE plpythonu TO user_name;  
GRANT USAGE ON LANGUAGE plpgsql TO user_name;  
GRANT SELECT ON ALL TABLES IN SCHEMA pg_catalog TO user_name;  
GRANT SELECT ON pg_class_info TO user_name;  
GRANT SELECT ON sys_serverless_usage TO user_name;  
GRANT SELECT ON pg_database_info TO user_name;  
GRANT SELECT ON pg_statistic TO user_name;
```

Pada contoh sebelumnya, ganti *user_name* dengan nama pengguna Anda. Kemudian, ganti *db_name* dengan nama database Amazon Redshift target Anda. Selanjutnya, ganti *schema_name* dengan nama skema Amazon Redshift Anda. Ulangi GRANT CREATE ON SCHEMA operasi untuk setiap skema target di mana Anda akan menerapkan kode yang dikonversi atau memigrasikan data. Akhirnya, ganti *your_password* dengan password yang aman.

Anda dapat menerapkan paket ekstensi pada basis data Amazon Redshift target Anda. Paket ekstensi adalah modul add-on yang mengemulasi fungsi basis data sumber yang diperlukan saat mengonversi objek ke Amazon Redshift. Untuk informasi selengkapnya, lihat [Menggunakan paket ekstensi dengan AWS Schema Conversion Tool](#).

Untuk operasi ini, AWS SCT perlu izin untuk mengakses bucket Amazon S3 atas nama Anda. Untuk memberikan izin ini, buat pengguna AWS Identity and Access Management (IAM) dengan kebijakan berikut.

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

```

```
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:ListBucket",
        "s3:PutObject"
    ],
    "Resource": [
        "arn:aws:s3:::aws-sct-*"
    ]
},
{
    "Effect": "Allow",
    "Action": [
        "s3:ListAllMyBuckets"
    ],
    "Resource": ""
}
]
```

Memilih strategi dan aturan pengoptimalan untuk digunakan AWS SCT

Untuk mengoptimalkan cara AWS Schema Conversion Tool mengonversi skema gudang data Anda, Anda dapat memilih strategi dan aturan yang Anda inginkan untuk digunakan alat tersebut. Setelah mengonversi skema Anda, dan meninjau kunci yang disarankan, Anda dapat menyesuaikan aturan atau mengubah strategi Anda untuk mendapatkan hasil yang Anda inginkan.

Untuk memilih strategi dan aturan pengoptimalan Anda

1. Pilih Pengaturan, lalu pilih Pengaturan Proyek. Kotak dialog Pengaturan proyek saat ini muncul.
2. Di panel kiri, pilih Strategi Optimasi. Strategi optimasi muncul di panel kanan dengan default yang dipilih.
3. Untuk Sektor Strategi, pilih strategi optimasi yang ingin Anda gunakan. Anda dapat memilih dari opsi berikut:
 - Gunakan metadata, abaikan informasi statistik — Dalam strategi ini, hanya informasi dari metadata yang digunakan untuk keputusan optimasi. Misalnya, jika ada lebih dari satu indeks pada tabel sumber, urutan pengurutan database sumber digunakan, dan indeks pertama menjadi kunci distribusi.

- Abaikan metadata, gunakan informasi statistik — Dalam strategi ini, keputusan pengoptimalan hanya berasal dari informasi statistik. Strategi ini hanya berlaku untuk tabel dan kolom yang statistiknya disediakan. Untuk informasi selengkapnya, lihat [Mengumpulkan atau mengunggah statistik untuk AWS SCT](#).
 - Gunakan metadata dan gunakan informasi statistik — Dalam strategi ini, metadata dan statistik digunakan untuk keputusan optimasi.
4. Setelah Anda memilih strategi optimasi Anda, Anda dapat memilih aturan yang ingin Anda gunakan. Anda dapat memilih dari opsi berikut:
- Pilih Distribution Key dan Sort Keys menggunakan metadata
 - Pilih tabel fakta dan dimensi yang sesuai untuk pemeriksaan
 - Menganalisis kardinalitas kolom indeks
 - Temukan tabel dan kolom yang paling sering digunakan dari tabel log kueri

Untuk setiap aturan, Anda dapat memasukkan bobot untuk kunci sortir dan bobot untuk kunci distribusi. AWS SCT menggunakan bobot yang Anda pilih saat mengubah skema Anda. Kemudian, ketika Anda meninjau kunci yang disarankan, jika Anda tidak puas dengan hasilnya, Anda dapat kembali ke sini dan mengubah pengaturan Anda. Untuk informasi selengkapnya, lihat [Mengelola dan menyesuaikan kunci di AWS SCT](#).

Mengumpulkan atau mengunggah statistik untuk AWS SCT

Untuk mengoptimalkan cara AWS Schema Conversion Tool mengonversi skema gudang data Anda, Anda dapat memberikan statistik dari basis data sumber yang dapat digunakan alat tersebut. Anda dapat mengumpulkan statistik langsung dari database, atau mengunggah file statistik yang ada.

Untuk menyediakan dan meninjau statistik

1. Buka proyek Anda dan sambungkan ke database sumber Anda.

Membuat aturan migrasi di AWS SCT

Sebelum mengonversi skema AWS SCT, Anda dapat mengatur aturan migrasi. Aturan migrasi dapat melakukan hal-hal seperti mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek. Misalnya, anggaplah Anda memiliki satu set tabel dalam skema sumber Anda bernama `test_TABLE_NAME`. Anda dapat mengatur aturan yang mengubah awalan `test_` ke awalan `demo_` dalam skema target.

Note

Anda dapat membuat aturan migrasi hanya untuk mesin basis data sumber dan target yang berbeda.

Anda dapat membuat aturan migrasi yang melakukan tugas-tugas berikut:

- Menambahkan, menghapus, atau mengganti awalan
- Menambahkan, menghapus, atau mengganti sufiks
- Ubah pemeriksaan kolom
- Ubah tipe data
- Mengubah panjang `char`, `varchar`, `nvarchar`, dan tipe `string` data
- Pindahkan objek
- Ganti nama objek

Anda dapat membuat aturan migrasi untuk objek berikut:

- Basis Data
- Skema
- Tabel
- Kolom

Membuat aturan migrasi

Anda dapat membuat aturan migrasi dan menyimpan aturan sebagai bagian dari proyek Anda. Dengan proyek Anda terbuka, gunakan prosedur berikut untuk membuat aturan migrasi.

Untuk membuat aturan migrasi

1. Pada menu Tampilan, pilih Tampilan pemetaan.
2. Dalam pemetaan Server, pilih sepasang server sumber dan target.
3. Pilih Aturan migrasi baru. Kotak dialog aturan Transformasi muncul.
4. Pilih Tambahkan aturan baru. Baris baru ditambahkan ke daftar aturan.
5. Konfigurasi aturan Anda:
 - a. Untuk Nama, masukkan nama untuk topik Anda.
 - b. Untuk Untuk, pilih jenis objek yang aturan berlaku.
 - c. Untuk tempat, masukkan filter untuk diterapkan ke objek sebelum menerapkan aturan migrasi. Klausula where dievaluasi dengan menggunakan klausa like. Anda dapat memasukkan nama yang tepat untuk memilih satu objek, atau Anda dapat memasukkan pola untuk memilih beberapa objek.

Bidang yang tersedia untuk klausa where berbeda tergantung pada jenis objek. Misalnya, jika jenis objek adalah skema hanya ada satu bidang yang tersedia, untuk nama skema.
 - d. Untuk Tindakan, pilih jenis aturan migrasi yang ingin Anda buat.
 - e. Tergantung pada jenis aturan, masukkan satu atau dua nilai tambahan. Misalnya, untuk mengganti nama objek, masukkan nama baru objek. Untuk mengganti awalan, masukkan awalan lama dan awalan baru.
6. Setelah mengonfigurasi aturan migrasi, pilih Simpan untuk menyimpan aturan Anda. Anda juga dapat memilih Batal untuk membatalkan perubahan Anda.

Transformation rules affect how the converted objects to be named on the target database. For example, you can rename a schema or table, add or remove prefixes or suffixes from object names, convert names to lowercase or uppercase, etc. When defining object names, it is possible to use % as a wildcard. The order in which the rules are applied can be defined using drag-and-drop. Rules lower in the list have a higher priority. Default transformation rules are always at the top of the list and can be disabled or changed only in the [Conversion settings](#) tab. The rules can be exported to a file for later use in the DMS, but please note that AWS DMS *doesn't support* more than one transformation rule per schema level or per table level. Note, every rule might have to following status along with the corresponding color:

- Successfully created enabled rule
- Rule with incorrect data entered

Transformation rule: For **tables** where database name is like '%' and schema name is like '%' and table name is like 'test_%' add prefix 'demo_%'

Name: Transformation rule

For: table

where database name like: % schema name like: % table name like: test_%

Actions: add prefix demo_%

Buttons: Save, Cancel, + Add new rule, Export script for DMS, Import script into SCT, Save all, Close

7. Setelah Anda selesai menambahkan, mengedit, dan menghapus aturan, pilih Simpan Semua untuk menyimpan semua perubahan Anda.
8. Pilih Tutup untuk menutup kotak dialog Aturan transformasi.

Anda dapat menggunakan ikon sakelar untuk menonaktifkan aturan migrasi tanpa menghapusnya. Anda dapat menggunakan ikon salin untuk menduplikasi aturan migrasi yang ada. Anda dapat menggunakan ikon pensil untuk mengedit aturan migrasi yang ada. Anda dapat menggunakan ikon hapus untuk menghapus aturan migrasi yang ada. Untuk menyimpan perubahan apa pun yang Anda buat pada aturan migrasi, pilih Simpan Semua.

Mengekspor aturan migrasi

Jika Anda menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan data dari basis data sumber ke basis data target, Anda dapat memberikan informasi tentang aturan migrasi ke AWS DMS. Untuk informasi selengkapnya tentang tugas, lihat [Bekerja dengan tugas AWS Database Migration Service replikasi](#).

Untuk mengekspor aturan migrasi

1. Dalam AWS Schema Conversion Tool, pilih Mapping View pada menu View.
2. Di Aturan migrasi, pilih aturan migrasi, lalu pilih Ubah aturan migrasi.
3. Pilih Ekspor skrip untuk AWS DMS.

4. Jelajahi lokasi tempat Anda ingin menyimpan skrip, lalu pilih Simpan. Aturan migrasi Anda disimpan sebagai JSON skrip yang dapat digunakan oleh AWS DMS.

Mengonversi skema Anda menggunakan AWS SCT

Setelah Anda menghubungkan proyek Anda ke database sumber dan database target Anda, AWS Schema Conversion Tool proyek Anda menampilkan skema dari database sumber Anda di panel kiri. Skema disajikan dalam format tampilan pohon, dan setiap simpul pohon dimuat dengan lambat. Ketika Anda memilih node dalam tampilan pohon, AWS SCT meminta informasi skema dari database sumber Anda pada saat itu.

Anda dapat memilih item skema dari basis data sumber Anda dan kemudian mengonversi skema menjadi skema setara untuk mesin database database target Anda. Anda dapat memilih item skema apa pun dari database sumber Anda untuk dikonversi. Jika item skema yang Anda pilih bergantung pada item induk, maka buat AWS SCT juga skema untuk item induk. Misalnya, jika Anda memilih kolom dari tabel untuk dikonversi AWS SCT, maka buat skema untuk kolom, tabel tempat kolom berada, dan database tempat tabel berada.

Mengubah skema

Untuk mengonversi skema dari database sumber Anda, pilih kotak centang untuk nama skema yang akan dikonversi. Selanjutnya, pilih skema ini dari panel kiri proyek Anda. AWS SCT menyoroti nama skema dengan warna biru. Buka menu konteks (klik kanan) untuk skema, dan pilih Konversi skema, seperti yang ditunjukkan berikut.

File Actions Main view Settings Applications Help Add source Add target

Connected. Click to disconnect

Servers

- SQL Server - ec2-52-17-21-76.eu-west-1.compute.am
 - Databases [12]
 - AdventureWorks2012_CS
 - alfresco
 - GOLD_TEST_SS_PG
 - LARGE_DB_SS
 - master
 - model
 - msdb
 - tempdb
 - TEST**
 - vmap
 - vpas
 - vrecon
 - Server Objects
 - SQL Server Agent
 - Applications
 - SQL Scripts
 - noSQL Clusters
 - ETL

Create mapping...
Create report
Convert schema
Register agent
Compare schema
Load schema
Hide schema
Refresh from database
Collect statistics
Upload statistics
Create DMS task
Create Local & DMS task
Create Local task
Add virtual partitioning
Save as SQL

Properties SQL Related converted objects Statistics

Name	
Created or last modified	
Created	2021-09-06 09:56:08.26
Object name	
Name	TEST
compatibility-level	100
collation-name	SQL_Latin1_General_CP1_CI_AS

Properties SQL Apply status Key management

Name	
Category	
Name	<Aurora_MySQL (virtual)>

Setelah Anda mengonversi skema dari database sumber Anda, Anda dapat memilih item skema dari panel kiri proyek Anda dan melihat skema yang dikonversi di panel tengah proyek Anda. Panel tengah bawah menampilkan properti dan SQL perintah untuk membuat skema yang dikonversi, seperti yang ditunjukkan berikut.

The screenshot displays the AWS Schema Conversion Tool interface. On the left, a tree view shows the server structure: SQL Server - ec2-52-17-21-76.eu-west-1.cc > Databases [12] > AdventureWorks2012_CS > alfresco > GOLD_TEST_SS_PG > LARGE_DB_SS > Schemas [2] > dbo > Tables [4] > Account. The main pane shows the SQL script for the source table:

```

1 CREATE TABLE [dbo].[Account] (
2 [ID] numeric(14,0) NOT NULL,
3 [AccountNo] varchar(16) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
4 [CurrencyID] numeric(3,0) NOT NULL,
5 [Description] varchar(160) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL,
6 [CustomerID] numeric(14,0) NOT NULL,
7 [StateID] numeric(2,0) NOT NULL,
8 [AccountBalance] numeric(14,3) NOT NULL,
9 [BlockedAmount] numeric(14,3) NOT NULL,
10 [Opendate] datetime NULL,
11 [Closedate] datetime NULL,
12 [RespManagerID] numeric(5,0) NULL,
13 [BankID] varchar(10) COLLATE SQL_Latin1_General_CP1_CI_AS NOT NULL
14 )
15 ON [PRIMARY];

```

Below the script, it indicates 'Cursor position: 0 Modified: true'. The bottom pane shows the target Amazon RDS instance configuration: 'Target Amazon RDS for MySQL table: Account'. The SQL script for the target instance is:

```

1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo VARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,

```

Setelah Anda mengubah skema Anda, Anda dapat menyimpan proyek Anda. Informasi skema dari database sumber Anda disimpan dengan proyek Anda. Fungsionalitas ini berarti Anda dapat bekerja offline tanpa terhubung ke database sumber Anda. AWS SCT menghubungkan ke database sumber Anda untuk memperbarui skema dalam proyek Anda jika Anda memilih Refresh from Database untuk database sumber Anda. Untuk informasi selengkapnya, lihat [Memperbarui dan menyegarkan skema konversi Anda di AWS SCT](#).

Anda dapat membuat laporan penilaian migrasi database dari item yang tidak dapat dikonversi secara otomatis. Laporan penilaian berguna untuk mengidentifikasi dan menyelesaikan item skema yang tidak dapat dikonversi secara otomatis. Untuk informasi selengkapnya, lihat [Menggunakan laporan penilaian di AWS Schema Conversion Tool](#).

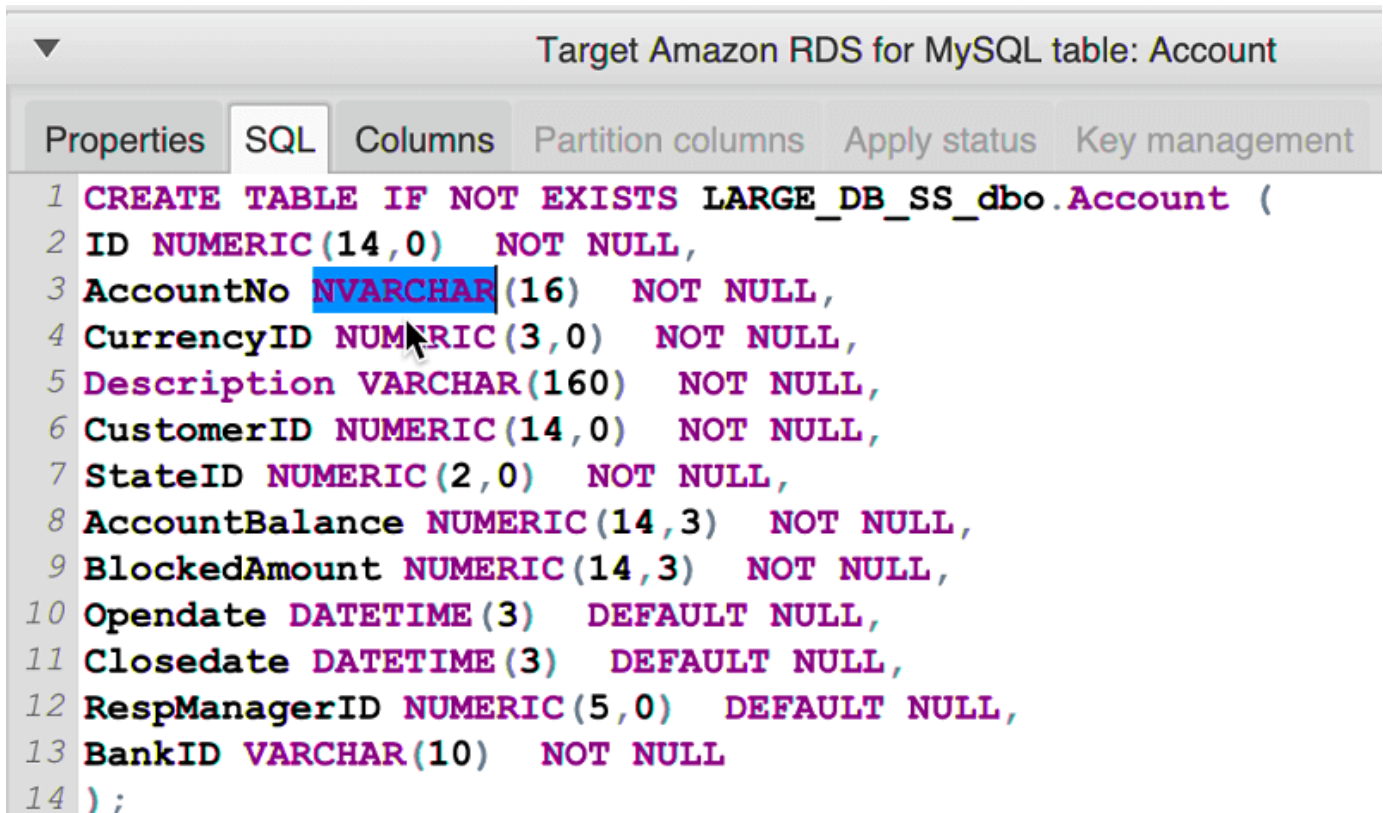
Saat AWS SCT menghasilkan skema yang dikonversi, itu tidak segera menerapkannya ke database target. Sebagai gantinya, skema yang dikonversi disimpan secara lokal sampai Anda siap menerapkannya ke database target. Untuk informasi selengkapnya, lihat [Menerapkan skema konversi Anda](#).

Mengedit skema yang dikonversi

Anda dapat mengedit skema yang dikonversi dan menyimpan perubahan sebagai bagian dari proyek Anda.

Untuk mengedit skema yang dikonversi

1. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih item skema yang ingin Anda edit skema yang dikonversi.
2. Di panel tengah bawah yang menampilkan skema yang dikonversi untuk item yang dipilih, pilih tab. SQL
3. Dalam teks yang ditampilkan untuk SQLtab, ubah skema sesuai kebutuhan. Skema secara otomatis disimpan dengan proyek Anda saat Anda memperbaruinya.



```
▼ Target Amazon RDS for MySQL table: Account
Properties SQL Columns Partition columns Apply status Key management
1 CREATE TABLE IF NOT EXISTS LARGE_DB_SS_dbo.Account (
2 ID NUMERIC(14,0) NOT NULL,
3 AccountNo NVARCHAR(16) NOT NULL,
4 CurrencyID NUMERIC(3,0) NOT NULL,
5 Description VARCHAR(160) NOT NULL,
6 CustomerID NUMERIC(14,0) NOT NULL,
7 StateID NUMERIC(2,0) NOT NULL,
8 AccountBalance NUMERIC(14,3) NOT NULL,
9 BlockedAmount NUMERIC(14,3) NOT NULL,
10 Opendate DATETIME(3) DEFAULT NULL,
11 Closedate DATETIME(3) DEFAULT NULL,
12 RespManagerID NUMERIC(5,0) DEFAULT NULL,
13 BankID VARCHAR(10) NOT NULL
14 );
```

Perubahan yang Anda buat pada skema yang dikonversi disimpan dengan proyek Anda saat Anda melakukan pembaruan. Jika Anda baru mengonversi item skema dari basis data sumber Anda, dan Anda telah membuat pembaruan ke skema yang dikonversi sebelumnya untuk item tersebut, pembaruan yang ada akan digantikan oleh item skema yang baru dikonversi berdasarkan basis data sumber Anda.

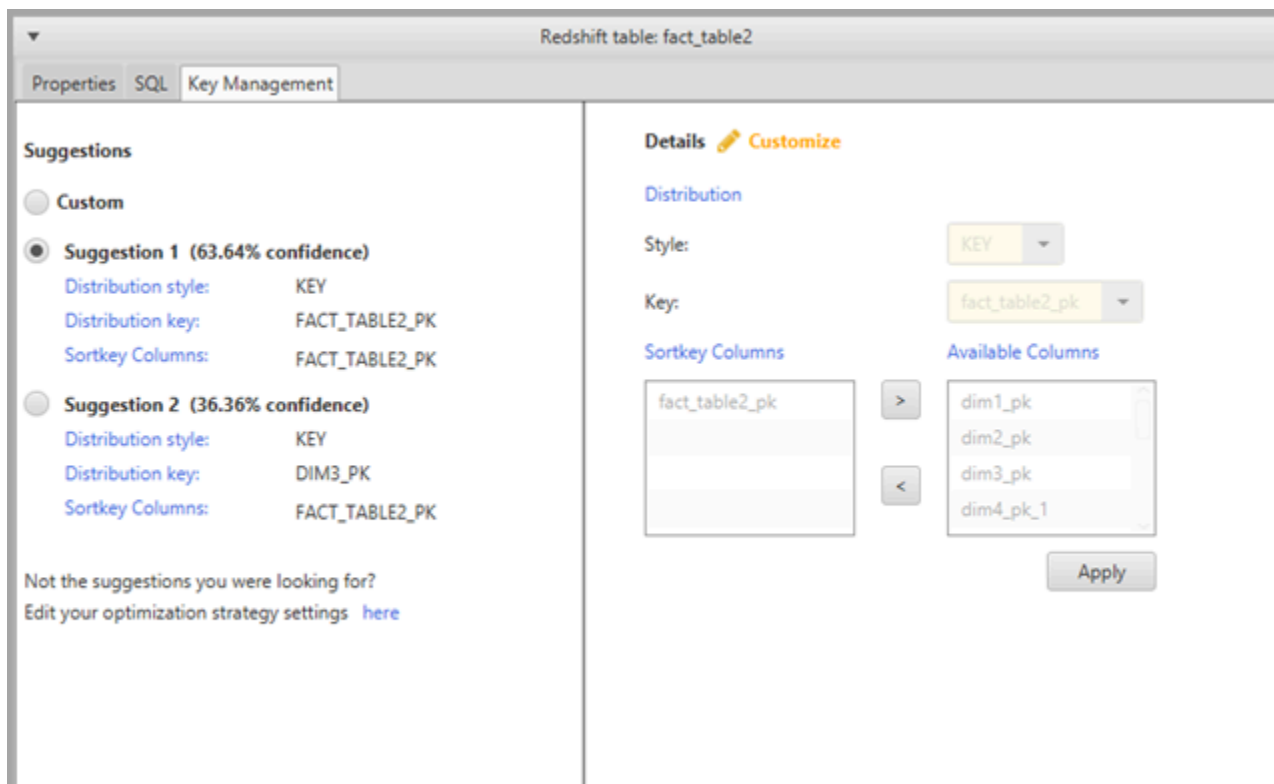
Membersihkan skema yang dikonversi

Sampai Anda menerapkan skema ke database target Anda, AWS SCT hanya menyimpan skema yang dikonversi secara lokal di proyek Anda. Anda dapat menghapus skema yang direncanakan dari proyek Anda dengan memilih node tampilan pohon untuk database target Anda, lalu memilih Refresh from Database. Karena tidak ada skema yang ditulis ke database target Anda, penyegaran dari database akan menghapus elemen skema yang direncanakan dalam AWS SCT proyek Anda agar sesuai dengan apa yang ada di database target Anda.

Mengelola dan menyesuaikan kunci di AWS SCT

Setelah Anda mengonversi skema Anda dengan AWS Schema Conversion Tool, Anda dapat mengelola dan mengedit kunci Anda. Manajemen kunci adalah jantung dari konversi gudang data.

Untuk mengelola kunci, pilih tabel di database target Anda, lalu pilih tab Manajemen Kunci seperti yang ditunjukkan berikut.



Panel kiri berisi saran utama, dan termasuk peringkat kepercayaan untuk setiap saran. Anda dapat memilih salah satu saran, atau Anda dapat menyesuaikan kunci dengan mengeditnya di panel kanan.

Jika pilihan untuk kunci tidak terlihat seperti yang Anda harapkan, Anda dapat mengedit strategi pengoptimalan edit Anda, dan kemudian mencoba lagi konversi. Untuk informasi selengkapnya, lihat [Memilih strategi dan aturan pengoptimalan untuk digunakan AWS SCT](#).

Topik terkait

- [Pilih tombol sortir terbaik](#)
- [Pilih gaya distribusi terbaik](#)

Membuat dan menggunakan laporan penilaian di AWS SCT

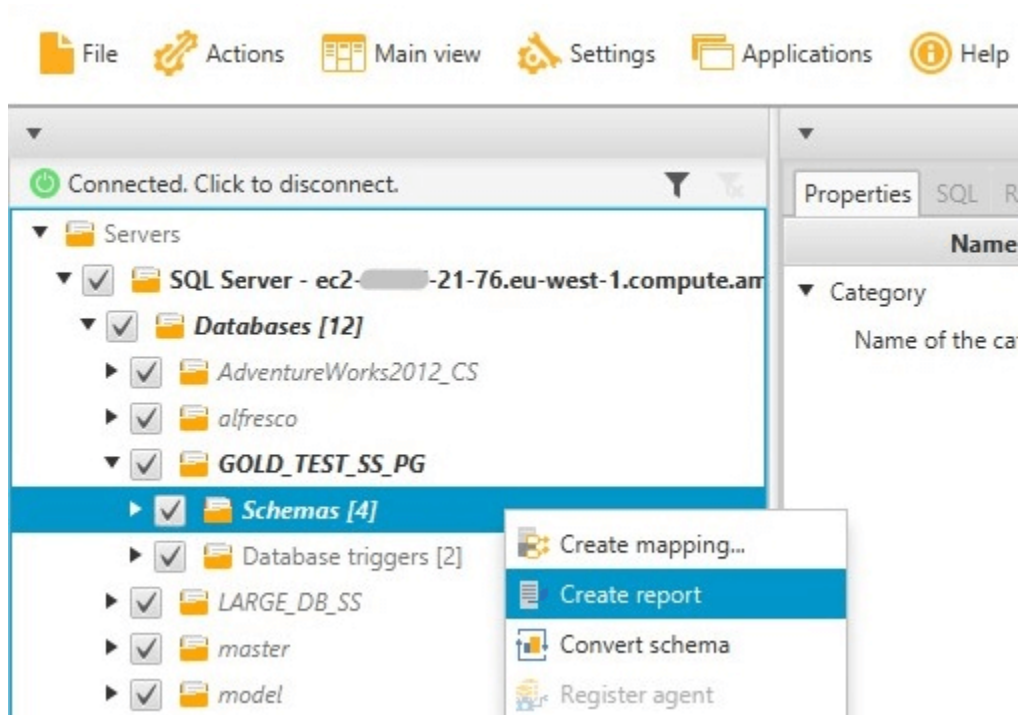
AWS Schema Conversion Tool Membuat laporan penilaian migrasi database untuk membantu Anda mengonversi skema. Laporan penilaian migrasi database memberikan informasi penting tentang konversi skema dari database sumber Anda ke database target Anda. Laporan ini merangkum semua tugas konversi skema dan merinci item tindakan untuk skema yang tidak dapat dikonversi ke mesin DB database target Anda. Laporan ini juga mencakup perkiraan jumlah upaya yang diperlukan untuk menulis kode yang setara dalam database target Anda yang tidak dapat dikonversi secara otomatis.

Membuat laporan penilaian migrasi database

Gunakan prosedur berikut untuk membuat laporan penilaian migrasi database.

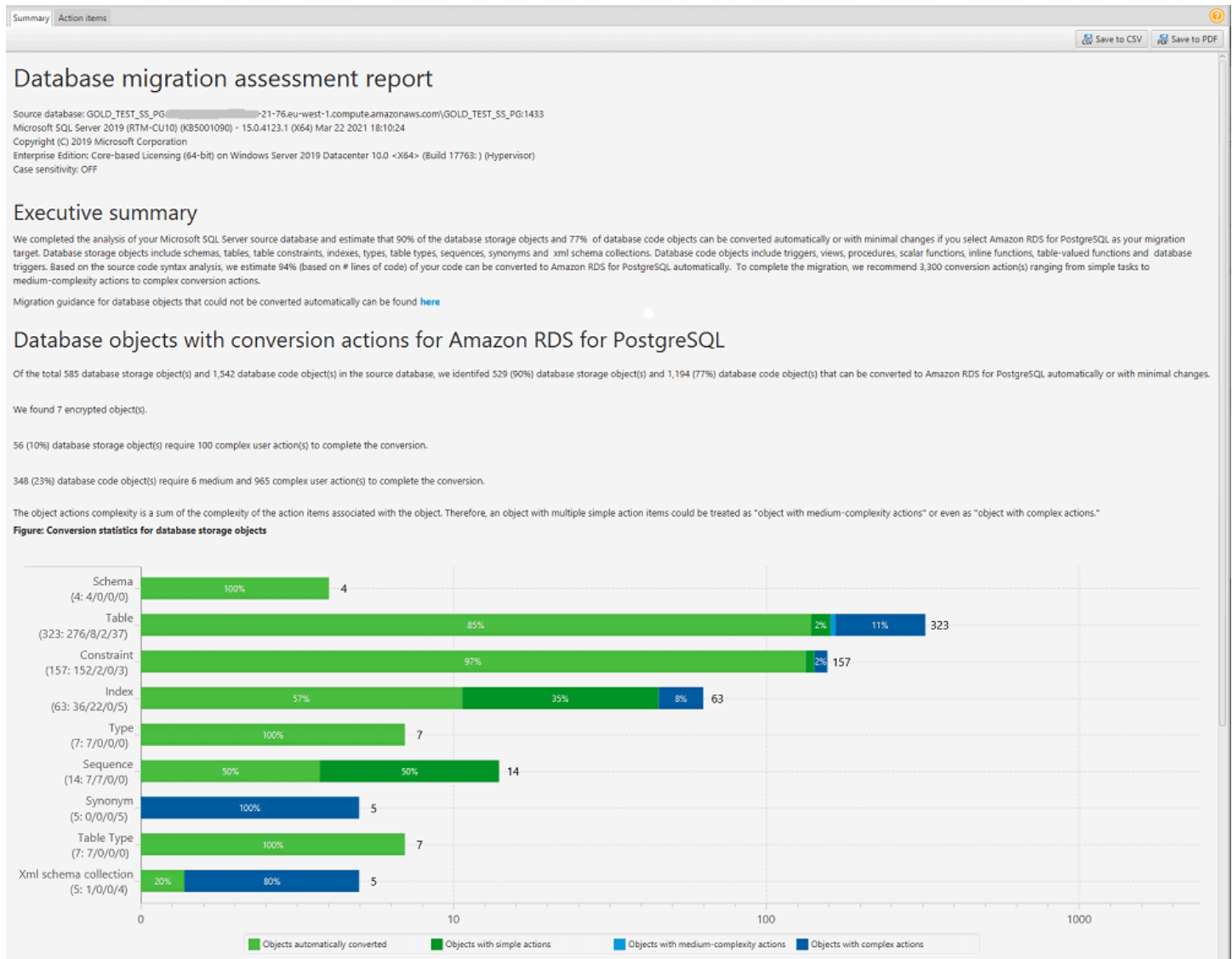
Untuk membuat laporan penilaian migrasi database

1. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih objek skema untuk membuat laporan penilaian.
2. Buka menu konteks (klik kanan) untuk objek, lalu pilih Buat Laporan.



Ringkasan laporan penilaian

Setelah Anda membuat laporan penilaian, tampilan laporan penilaian akan terbuka, menampilkan tab Ringkasan. Tab Ringkasan menampilkan informasi ringkasan dari laporan penilaian migrasi database. Ini menunjukkan item yang dikonversi secara otomatis, dan item yang tidak dikonversi secara otomatis.



Untuk item skema yang tidak dapat dikonversi secara otomatis ke mesin basis data target, ringkasan mencakup perkiraan upaya yang diperlukan untuk membuat item skema dalam instans DB target Anda yang setara dengan yang ada di sumber Anda.

Laporan tersebut mengkategorikan perkiraan waktu untuk mengonversi item skema ini sebagai berikut:

- **Simple** — Tindakan yang dapat diselesaikan dalam waktu kurang dari satu jam.
- **Medium** — Tindakan yang lebih kompleks dan dapat diselesaikan dalam satu hingga empat jam.
- **Signifikan** — Tindakan yang sangat kompleks dan membutuhkan waktu lebih dari empat jam untuk menyelesaikannya.

Item tindakan laporan penilaian

Tampilan laporan penilaian juga menyertakan tab Item Tindakan. Tab ini berisi daftar item yang tidak dapat dikonversi secara otomatis ke mesin database database target Anda. Jika Anda memilih item tindakan dari daftar, AWS SCT sorot item dari skema Anda yang berlaku untuk item tindakan.

Laporan ini juga berisi rekomendasi tentang cara mengonversi item skema secara manual. Untuk informasi selengkapnya tentang memutuskan cara menangani konversi manual, lihat [Menangani konversi manual di AWS SCT](#).

The screenshot shows the AWS Schema Conversion Tool interface. The top menu includes File, Actions, Assessment Report view, Settings, Applications, Help, Add source, and Add target. The main window is divided into several sections:

- Summary / Action items:** Shows a tree view of servers and databases. The selected server is 'SQL Server - ec2-21-76.eu-west-1.compute'. Underneath, there are databases like 'AdventureWorks2012_CS', 'alfresco', 'GOLD_TEST_SS_PG', 'LARGE_DB_SS', 'master', 'model', 'msdb', 'tempdb', and 'TEST'. There is also a 'Schemas [1]' section with a 'dbo' schema containing various tables, views, and procedures.
- Group by / Complexity / Object type:** Filters for 'Issues', 'All' complexity, and 'All' object types.
- Action Items List:**
 - Issue 609:** MySQL doesn't support the OUTPUT clause in the statements INSERT, UPDATE, and DELETE. A manual conversion is required. Recommended action: Create a trigger for INSERT statements for the table, and then save the inserted rows in a temporary table. After the INSERT operation, you can make use of the rows saved in the temporary table. Number of occurrences: 1 | Documentation reference(s): <http://dev.mysql.com/doc/refman/8.0/en/insert.html>
 - Issue 681:** MySQL doesn't support creating indexes with a CLUSTER option. The user can't create CLUSTER INDEX, MySQL will create it automatically. Recommended action: Use non-clustered indexes. Number of occurrences: 2
 - Issue 794:** MySQL doesn't support user-defined data types. The user datatype has been replaced by the base datatype. Recommended action: Please review generated code and modify it if necessary. Number of occurrences: 1. Parameter: @InputPosNo (Number of occurrences: 1).
 - Issue 826:** Check the default value for a DateTime variable. Recommended action: Check the default value for a DateTime variable. Number of occurrences: 1.
 - Issue 844:** MySQL expands fractional seconds support for TIME, DATETIME2 and DATETIMEOFFSET values, with up to microseconds (6 digits) of precision. Recommended action: Review your transformed code and modify it if necessary to avoid a loss of accuracy. Number of occurrences: 8 | Documentation reference(s): <https://dev.mysql.com/doc/refman/8.0/en/data-types.html>
 - Issue 9997:** Unable to resolve objects. Recommended action: Verify if the unresolved object is present in the database. If it isn't, check the object name or add the object. If the object is present, transform the code manually. Number of occurrences: 3
 - Issue 690:** MySQL doesn't support table types. Recommended action: Perform a manual conversion. Number of occurrences: 1
 - Issue 811:** Unable to convert functions. Recommended action: Create a user-defined function. Number of occurrences: 12
- Source Microsoft SQL Server procedure: POSITION_UPDATE_CASH_CGT_BULK:**

```

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

```
- Target Amazon RDS for MySQL category: Schemas:**

Name	Value
Category	Schemas
Name of the category	Schemas

Menyimpan laporan penilaian

Anda dapat menyimpan salinan lokal laporan penilaian migrasi database sebagai PDF file atau file values (CSV) yang dipisahkan koma. CSVFile hanya berisi informasi item tindakan. PDFFile berisi informasi ringkasan dan item tindakan, seperti yang ditunjukkan pada contoh berikut.

Database objects with conversion actions for Amazon RDS for PostgreSQL

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

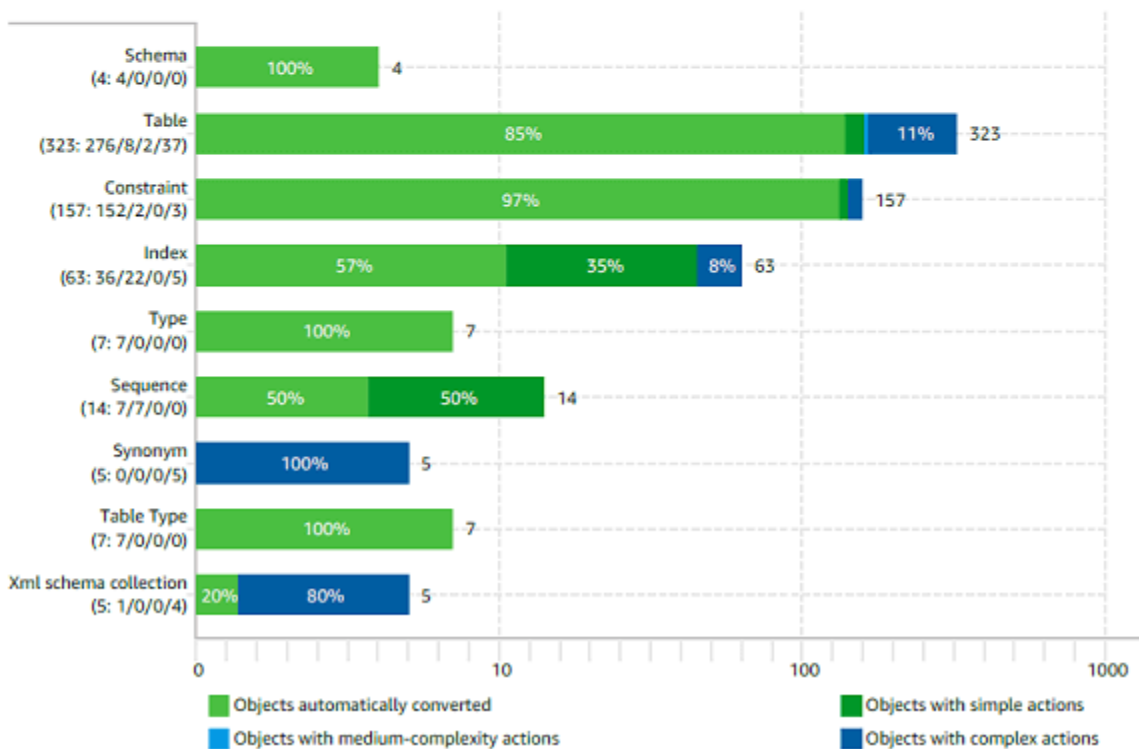
We found 7 encrypted object(s).

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

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

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

Figure: Conversion statistics for database storage objects



Menangani konversi manual di AWS SCT

Laporan penilaian mencakup daftar item yang tidak dapat dikonversi secara otomatis ke mesin database database target Anda. Untuk setiap item yang tidak dapat dikonversi, ada item tindakan di Item Tindakan tab.

Anda dapat menanggapi item tindakan dalam laporan penilaian dengan cara berikut:

- Ubah skema basis data sumber Anda.

- Ubah skema basis data target Anda.

Memodifikasi skema sumber Anda

Untuk beberapa item, mungkin lebih mudah untuk memodifikasi skema database di database sumber Anda ke skema yang dapat dikonversi secara otomatis. Pertama, verifikasi bahwa perubahan baru kompatibel dengan arsitektur aplikasi Anda, lalu perbarui skema di database sumber Anda. Terakhir, segarkan proyek Anda dengan informasi skema yang diperbarui. Anda kemudian dapat mengonversi skema yang diperbarui, dan membuat laporan penilaian migrasi database baru. Item tindakan tidak lagi muncul untuk item yang berubah dalam skema sumber.

Keuntungan dari proses ini adalah skema Anda yang diperbarui selalu tersedia saat Anda menyegarkan dari basis data sumber Anda.

Memodifikasi skema target Anda

Untuk beberapa item, mungkin lebih mudah untuk menerapkan skema yang dikonversi ke database target Anda, dan kemudian menambahkan item skema yang setara secara manual ke database target Anda untuk item yang tidak dapat dikonversi secara otomatis. Anda dapat menulis semua skema yang dapat dikonversi secara otomatis ke database target Anda dengan menerapkan skema. Untuk informasi selengkapnya, lihat [Menyimpan dan menerapkan skema konversi Anda di AWS SCT](#).

Skema yang ditulis ke database target Anda tidak berisi item yang tidak dapat dikonversi secara otomatis. Setelah menerapkan skema ke database target Anda, Anda kemudian dapat secara manual membuat skema dalam database target Anda yang setara dengan yang ada di database sumber. Item tindakan dalam laporan penilaian migrasi database berisi saran tentang cara membuat skema yang setara.

Warning

Jika Anda membuat skema secara manual di database target Anda, simpan salinan pekerjaan manual apa pun yang Anda lakukan. Jika Anda menerapkan skema yang dikonversi dari proyek Anda ke database target Anda lagi, itu menimpa pekerjaan manual yang telah Anda lakukan.

Dalam beberapa kasus, Anda tidak dapat membuat skema yang setara dalam database target Anda. Anda mungkin perlu merancang ulang sebagian dari aplikasi dan database Anda untuk

menggunakan fungsionalitas yang tersedia dari mesin untuk basis data target Anda. Dalam kasus lain, Anda dapat mengabaikan skema yang tidak dapat dikonversi secara otomatis.

Memperbarui dan menyegarkan skema konversi Anda di AWS SCT

Anda dapat memperbarui skema sumber dan skema target dalam proyek Anda AWS Schema Conversion Tool .

- Sumber - Jika Anda memperbarui skema untuk database sumber Anda, AWS SCT ganti skema dalam proyek Anda dengan skema terbaru dari database sumber Anda. Dengan menggunakan fungsi ini, Anda dapat memperbarui proyek Anda jika perubahan telah dilakukan pada skema database sumber Anda.
- Target - Jika Anda memperbarui skema untuk database target Anda, AWS SCT ganti skema dalam proyek Anda dengan skema terbaru dari database target Anda. Jika Anda belum menerapkan skema apa pun ke database target Anda, AWS SCT hapus skema yang dikonversi dari proyek Anda. Anda kemudian dapat mengonversi skema dari database sumber Anda untuk database target yang bersih.

Anda memperbarui skema dalam AWS SCT proyek Anda dengan memilih Refresh dari database.

Menyimpan dan menerapkan skema konversi Anda di AWS SCT

Saat AWS Schema Conversion Tool menghasilkan skema yang dikonversi (seperti yang ditunjukkan pada [Mengonversi skema Anda menggunakan AWS SCT](#)), skema tersebut tidak segera menerapkan skema yang dikonversi ke database target. Sebagai gantinya, skema yang dikonversi disimpan secara lokal di proyek Anda sampai Anda siap menerapkannya ke database target. Dengan menggunakan fungsi ini, Anda dapat bekerja dengan item skema yang tidak dapat dikonversi secara otomatis ke mesin basis data target Anda. Untuk informasi selengkapnya tentang item yang tidak dapat dikonversi secara otomatis, lihat [Menggunakan laporan penilaian di AWS Schema Conversion Tool](#).

Anda dapat secara opsional meminta alat menyimpan skema yang dikonversi ke file sebagai SQL skrip sebelum menerapkan skema ke database target Anda. Anda juga dapat meminta alat menerapkan skema yang dikonversi langsung ke database target Anda.

Menyimpan skema konversi Anda ke file

Anda dapat menyimpan skema yang dikonversi sebagai SQL skrip dalam file teks. Dengan menggunakan pendekatan ini, Anda dapat memodifikasi SQL skrip yang dihasilkan dari AWS SCT ke item alamat yang alat tidak dapat dikonversi secara otomatis. Anda kemudian dapat menjalankan skrip yang diperbarui pada instans DB target Anda untuk menerapkan skema yang dikonversi ke database target Anda.

Untuk menyimpan skema yang dikonversi sebagai skrip SQL

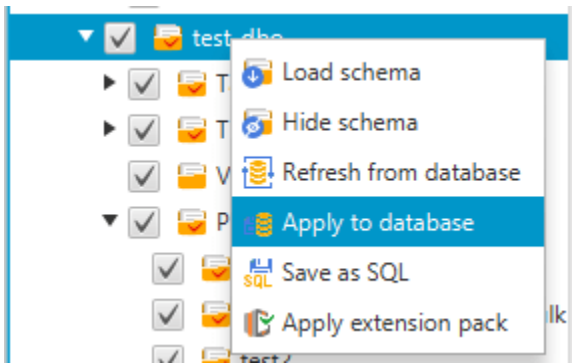
1. Pilih skema Anda dan buka menu konteks (klik kanan).
2. Pilih Simpan sebagai SQL.
3. Masukkan nama file dan pilih Simpan.
4. Simpan skema Anda yang telah dikonversi menggunakan salah satu opsi berikut:
 - File tunggal
 - File tunggal per tahap
 - File tunggal per pernyataan

Untuk memilih format SQL skrip

1. Pada menu Pengaturan, pilih Pengaturan proyek.
2. Pilih Simpan skrip.
3. Untuk Vendor, pilih platform database.
4. Untuk Menyimpan SQL skrip, pilih bagaimana Anda ingin menyimpan skrip skema database Anda.
5. Pilih OK untuk menyimpan pengaturan.

Menerapkan skema konversi Anda

Ketika Anda siap untuk menerapkan skema yang dikonversi ke database target Anda, pilih elemen skema dari panel kanan proyek Anda. Buka menu konteks (klik kanan) untuk elemen skema, lalu pilih Terapkan ke database, seperti yang ditunjukkan berikut.



Skema paket ekstensi

Pertama kali Anda menerapkan skema yang dikonversi ke instans DB target Anda, AWS SCT menambahkan skema tambahan ke instans DB target Anda. Skema ini mengimplementasikan fungsi sistem dari database sumber yang diperlukan saat menulis skema yang dikonversi ke instans DB target Anda. Skema ini disebut skema paket ekstensi.

Jangan mengubah skema paket ekstensi, atau Anda mungkin menemukan hasil yang tidak terduga dalam skema konversi yang ditulis ke instans DB target Anda. Ketika skema Anda sepenuhnya dimigrasikan ke instans DB target Anda, dan Anda tidak perlu lagi AWS SCT, Anda dapat menghapus skema paket ekstensi.

Skema paket ekstensi diberi nama sesuai dengan basis data sumber Anda sebagai berikut:

- Greenplum: `aws_greenplum_ext`
- SQLServer Microsoft: `aws_sqlserver_ext`
- Netezza: `aws_netezza_ext`
- Oracle: `aws_oracle_ext`
- Kepingan salju: `aws_snowflake_ext`
- Teradata: `aws_teradata_ext`
- Vertika: `aws_vertica_ext`

Untuk informasi selengkapnya, lihat [Menggunakan paket ekstensi dengan AWS Schema Conversion Tool](#).

Pustaka Python

Untuk membuat fungsi kustom di Amazon Redshift, Anda menggunakan bahasa Python. Gunakan paket AWS SCT ekstensi untuk menginstal pustaka python untuk database Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Menggunakan paket ekstensi dengan AWS Schema Conversion Tool](#).

Mengonversi data dari Amazon Redshift menggunakan AWS Schema Conversion Tool

Anda dapat menggunakan AWS Schema Conversion Tool untuk mengoptimalkan database Amazon Redshift Anda. Menggunakan database Amazon Redshift Anda sebagai sumber, dan menguji database Amazon Redshift sebagai target AWS SCT, merekomendasikan kunci pengurutan dan kunci distribusi untuk mengoptimalkan database Anda.

Mengoptimalkan database Amazon Redshift Anda

Gunakan prosedur berikut untuk mengoptimalkan database Amazon Redshift Anda.

Untuk mengoptimalkan database Amazon Redshift Anda

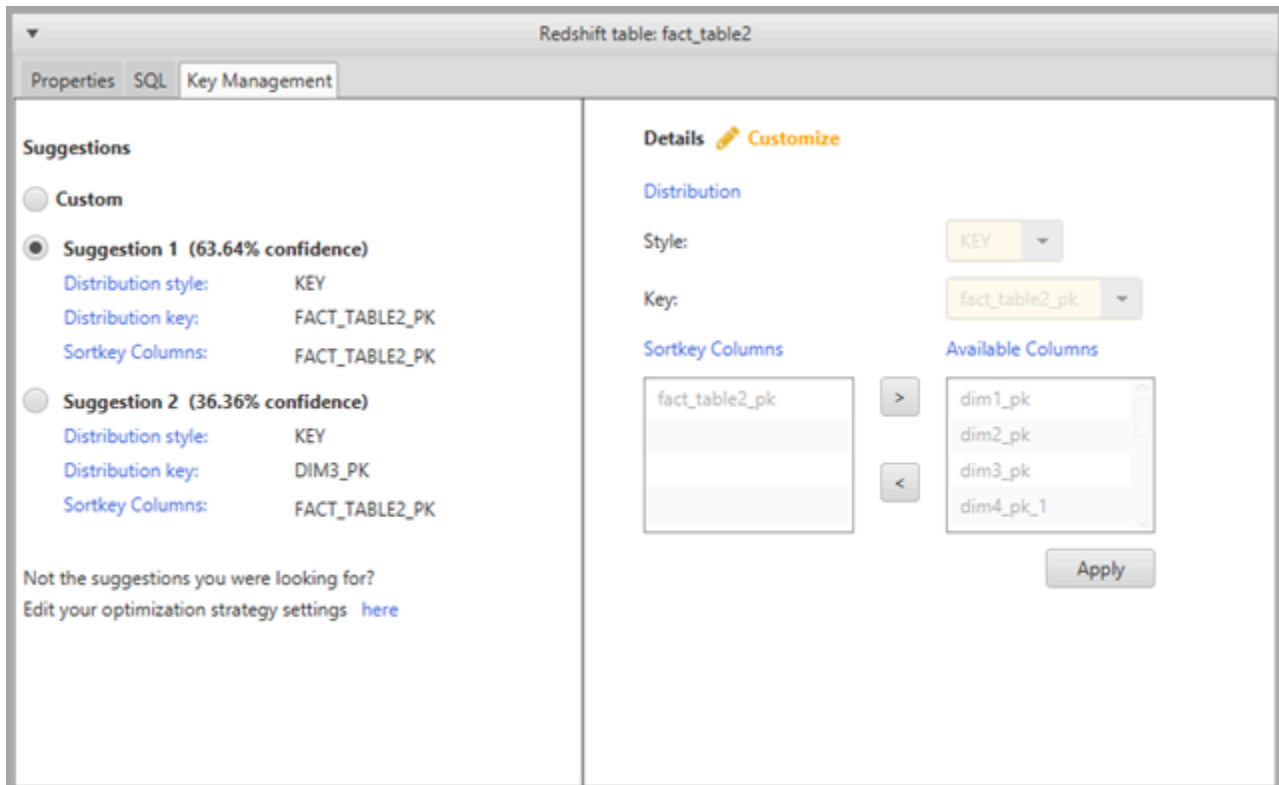
1. Ambil snapshot manual cluster Amazon Redshift Anda sebagai cadangan. Anda dapat menghapus snapshot setelah Anda selesai mengoptimalkan cluster Amazon Redshift Anda dan menguji setiap perubahan yang Anda buat. Untuk informasi selengkapnya, lihat [snapshot Amazon Redshift](#).
2. Pilih objek skema untuk dikonversi dari panel kiri proyek Anda. Buka menu konteks (klik kanan) untuk objek, lalu pilih Kumpulkan Statistik.

AWS SCT menggunakan statistik untuk membuat saran untuk kunci pengurutan dan kunci distribusi.

3. Pilih objek skema untuk dioptimalkan dari panel kiri proyek Anda. Buka menu konteks (klik kanan) untuk objek, lalu pilih Jalankan Optimasi.

AWS SCT membuat saran untuk kunci pengurutan dan kunci distribusi.

4. Untuk meninjau saran, perluas node tabel di bawah skema Anda di panel kiri proyek Anda, lalu pilih tabel. Pilih tab Manajemen Kunci seperti yang ditunjukkan berikut.



Panel kiri berisi saran utama, dan termasuk peringkat kepercayaan untuk setiap saran. Anda dapat memilih salah satu saran, atau Anda dapat menyesuaikan kunci dengan mengeditnya di panel kanan.

5. Anda dapat membuat laporan yang berisi saran pengoptimalan. Untuk membuat laporan, lakukan hal berikut:
 - a. Pilih objek skema yang Anda optimalkan dari panel kiri proyek Anda. Buka menu konteks (klik kanan) untuk objek, lalu pilih Buat Laporan.

Laporan terbuka di jendela utama, dan tab Ringkasan muncul. Jumlah objek dengan saran pengoptimalan muncul dalam laporan.
 - b. Pilih tab Item Tindakan untuk melihat saran utama dalam format laporan.
 - c. Anda dapat menyimpan salinan lokal laporan pengoptimalan sebagai PDF file atau file values (CSV) yang dipisahkan koma. CSVFile hanya berisi informasi item tindakan. PDFFile tersebut berisi informasi ringkasan dan item tindakan.
6. Untuk menerapkan pengoptimalan yang disarankan ke database Anda, pilih objek di panel kanan proyek Anda. Buka menu konteks (klik kanan) untuk objek, lalu pilih Terapkan ke database.

Mengonversi Data Menggunakan ETL Proses di AWS Schema Conversion Tool

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk memigrasikan proses extract, transform, dan load (ETL). Jenis migrasi ini mencakup konversi logika bisnis ETL terkait. Logika ini dapat berada baik di dalam gudang data sumber Anda atau dalam skrip eksternal yang Anda jalankan secara terpisah.

Saat ini, AWS SCT mendukung konversi ETL skrip ke objek ke AWS Glue dan Amazon RSQL Redshift, seperti yang ditunjukkan pada tabel berikut.

Sumber	Target
Skrip informatika ETL	Informatika
ETLPaket Layanan Integrasi Microsoft SQL Server (SSIS)	AWS Glue atau AWS Glue Studio
Skrip shell dengan perintah tertanam dari Teradata Basic Teradata Query () BTEQ	Pergeseran Merah Amazon RSQL
Skrip Teradata BTEQ ETL	AWS Glue atau Amazon Redshift RSQL
Skrip pekerjaan Teradata FastExport	Pergeseran Merah Amazon RSQL
Skrip pekerjaan Teradata FastLoad	Pergeseran Merah Amazon RSQL
Skrip pekerjaan Teradata MultiLoad	Pergeseran Merah Amazon RSQL

Topik

- [Mengonversi ETL proses ke dalam AWS GlueAWS Schema Conversion Tool](#)
- [Mengonversi ETL proses ke dalam AWS GlueAWS Schema Conversion Tool](#)
- [Mengonversi skrip Informatica ETL dengan AWS Schema Conversion Tool](#)
- [Konversi ke SSIS dengan AWS GlueAWS SCT](#)
- [Mengonversi SSIS paket ke dengan AWS Glue StudioAWS Schema Conversion Tool](#)

- [Mengonversi BTEQ skrip Teradata ke Amazon Redshift dengan RSQL AWS SCT](#)
- [Mengonversi skrip shell dengan BTEQ perintah tertanam ke Amazon Redshift dengan RSQL AWS Schema Conversion Tool](#)
- [Mengonversi FastExport skrip ke Amazon Redshift dengan RSQL AWS Schema Conversion Tool](#)
- [Mengonversi skrip FastLoad pekerjaan ke Amazon Redshift dengan RSQL AWS Schema Conversion Tool](#)
- [Mengonversi MultiLoad skrip ke Amazon Redshift dengan RSQL AWS Schema Conversion Tool](#)

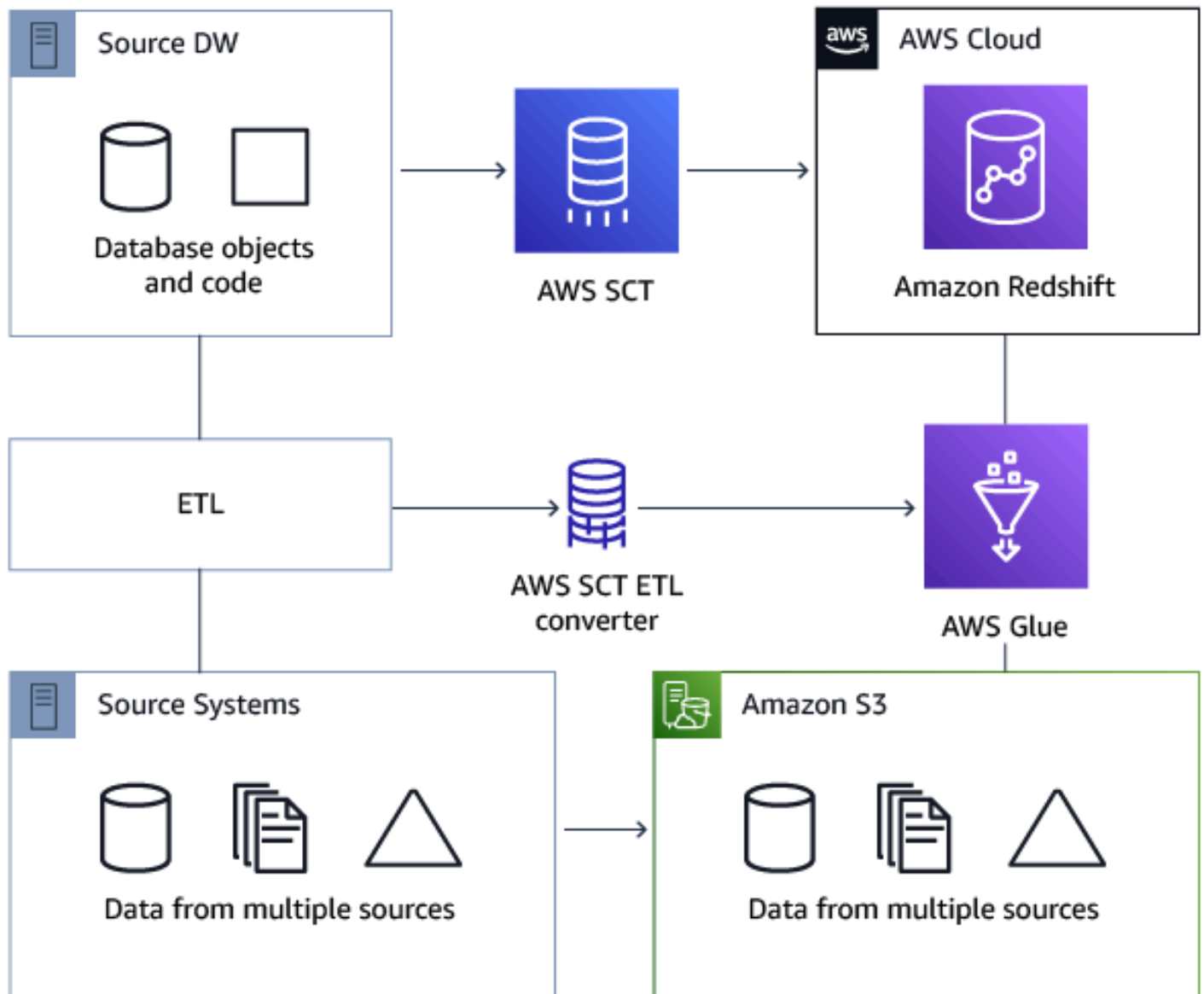
Mengonversi ETL proses ke dalam AWS GlueAWS Schema Conversion Tool

Berikut ini, Anda dapat menemukan garis besar proses untuk mengonversi ETL skrip ke AWS Glue with. AWS SCT Untuk contoh ini, kami mengonversi database Oracle ke Amazon Redshift, bersama dengan proses ETL yang digunakan dengan database sumber dan gudang data.

Topik

- [Prasyarat](#)
- [Memahami Katalog AWS Glue Data](#)
- [Batasan untuk mengonversi menggunakan dengan AWS SCTAWS Glue](#)
- [Langkah 1: Buat proyek baru](#)
- [Langkah 2: Buat AWS Glue pekerjaan](#)

Diagram arsitektur berikut menunjukkan contoh proyek migrasi database yang mencakup konversi ETL skrip ke AWS Glue.



Prasyarat

Sebelum memulai, lakukan hal berikut:

- Migrasikan database sumber apa pun yang ingin Anda migrasikan. AWS
- Migrasikan gudang data target ke. AWS
- Kumpulkan daftar semua kode yang terlibat dalam ETL proses Anda.
- Kumpulkan daftar semua informasi koneksi yang diperlukan untuk setiap database.

Selain itu, AWS Glue perlu izin untuk mengakses AWS sumber daya lain atas nama Anda. Anda memberikan izin tersebut dengan menggunakan AWS Identity and Access Management (IAM). Pastikan Anda membuat IAM kebijakan untuk AWS Glue. Untuk informasi selengkapnya, lihat [Membuat IAM kebijakan untuk AWS Glue service di Panduan Pengembang](#). AWS Glue

Memahami Katalog AWS Glue Data

Sebagai bagian dari proses konversi, AWS Glue memuat informasi mengenai sumber dan basis data target. Ini mengatur informasi ini ke dalam kategori, dalam struktur yang disebut pohon. Strukturnya meliputi:

- Koneksi - Parameter koneksi
- Crawler — Daftar crawler, satu crawler untuk setiap skema
- Database — Wadah yang menampung tabel
- Tabel — Definisi metadata yang mewakili data dalam tabel
- ETL pekerjaan — Logika bisnis yang melakukan ETL pekerjaan
- Pemicu — Logika yang mengontrol saat ETL pekerjaan berjalan AWS Glue (baik sesuai permintaan, berdasarkan jadwal, atau dipicu oleh peristiwa pekerjaan)

Katalog AWS Glue Data adalah indeks lokasi, skema, dan metrik runtime data Anda. Saat Anda bekerja dengan AWS Glue dan AWS SCT, Katalog AWS Glue Data berisi referensi ke data yang digunakan sebagai sumber dan target ETL pekerjaan Anda AWS Glue. Untuk membuat gudang data Anda, katalogkan data ini.

Anda menggunakan informasi dalam Katalog Data untuk membuat dan memantau ETL pekerjaan Anda. Biasanya, Anda menjalankan crawler untuk mengambil inventaris data di penyimpanan data Anda, tetapi ada cara lain untuk menambahkan tabel metadata ke dalam Katalog Data Anda.

Saat Anda menentukan tabel di Katalog Data, Anda menambahkannya ke database. Database digunakan untuk mengatur tabel di AWS Glue.

Batasan untuk mengonversi menggunakan dengan AWS SCT AWS Glue

Batasan berikut berlaku saat mengonversi menggunakan AWS SCT dengan AWS Glue.

Sumber	Batas default
--------	---------------

Jumlah database untuk setiap akun	10.000
Jumlah tabel untuk setiap database	100.000
Jumlah partisi untuk setiap tabel	1.000.000
Jumlah versi tabel untuk setiap tabel	100.000
Jumlah tabel untuk setiap akun	1.000.000
Jumlah partisi untuk setiap akun	10.000.000
Jumlah versi tabel untuk setiap akun	1.000.000
Jumlah koneksi untuk setiap akun	1.000
Jumlah crawler untuk setiap akun	25
Jumlah pekerjaan untuk setiap akun	25
Jumlah pemicu untuk setiap akun	25
Jumlah pekerjaan bersamaan yang dijalankan untuk setiap akun	30
Jumlah pekerjaan bersamaan yang dijalankan untuk setiap pekerjaan	3
Jumlah pekerjaan untuk setiap pemicu	10
Jumlah titik akhir pengembangan untuk setiap akun	5
Unit pemrosesan data maksimum (DPUs) yang digunakan oleh titik akhir pengembangan pada satu waktu	5
Maksimal DPUs digunakan oleh peran pada satu waktu	100

Panjang nama basis data	Tidak terbatas. Untuk kompatibilitas dengan toko metadata lainnya, seperti Apache Hive, namanya diubah untuk menggunakan karakter huruf kecil. Jika Anda berencana untuk mengakses database dari Amazon Athena, berikan nama dengan hanya karakter alfanumerik dan garis bawah.
Panjang nama koneksi	Tidak terbatas.
Panjang nama crawler	Tidak terbatas.

Langkah 1: Buat proyek baru

Untuk membuat proyek baru, ambil langkah-langkah tingkat tinggi ini:

1. Buat proyek baru di AWS SCT. Untuk informasi selengkapnya, lihat [Memulai dan mengelola Proyek di AWS SCT](#).
2. Tambahkan basis data sumber dan target Anda ke proyek. Untuk informasi selengkapnya, lihat [Menambahkan server ke proyek di AWS SCT](#).

Pastikan bahwa Anda telah memilih Gunakan AWS Glue dalam pengaturan koneksi database target. Untuk melakukannya, pilih AWS Glue. Untuk Salin dari AWS profil, pilih profil yang ingin Anda gunakan. Profil harus secara otomatis mengisi kunci AWS akses, kunci rahasia, dan folder bucket Amazon S3. Jika tidak, masukkan informasi ini sendiri. Setelah Anda memilih OK, AWS Glue menganalisis objek dan memuat metadata ke dalam Katalog Data. AWS Glue

Bergantung pada pengaturan keamanan Anda, Anda mungkin mendapatkan pesan peringatan yang mengatakan akun Anda tidak memiliki hak istimewa yang memadai untuk beberapa skema di server. Jika Anda memiliki akses ke skema yang Anda gunakan, Anda dapat mengabaikan pesan ini dengan aman.

3. Untuk menyelesaikan persiapan mengimpor Anda ETL, sambungkan ke basis data sumber dan target Anda. Untuk melakukannya, pilih database Anda di pohon metadata sumber atau target, lalu pilih Connect to the server.

AWS Glue membuat database di server database sumber dan satu di server database target untuk membantu ETL konversi. Database pada server target berisi Katalog AWS Glue Data. Untuk menemukan objek tertentu, gunakan pencarian pada sumber atau panel target.

Untuk melihat bagaimana objek tertentu mengonversi, cari item yang ingin Anda konversi, dan pilih Konversi skema dari menu konteksnya (klik kanan). AWS SCT mengubah objek yang dipilih ini menjadi skrip.

Anda dapat meninjau skrip yang dikonversi dari folder Scripts di panel kanan. Saat ini, skrip adalah objek virtual, yang hanya tersedia sebagai bagian dari AWS SCT proyek Anda.

Untuk membuat AWS Glue pekerjaan dengan skrip yang dikonversi, unggah skrip Anda ke Amazon S3. Untuk mengunggah skrip ke Amazon S3, pilih skrip, lalu pilih Simpan ke S3 dari menu konteksnya (klik kanan).

Langkah 2: Buat AWS Glue pekerjaan

Setelah Anda menyimpan skrip ke Amazon S3, Anda dapat memilihnya dan kemudian memilih Configure AWS Glue Job untuk membuka wizard untuk mengonfigurasi pekerjaan. AWS Glue Wizard membuatnya lebih mudah untuk mengatur ini:

1. Pada tab pertama wizard, Design Data Flow, Anda dapat memilih strategi eksekusi dan daftar skrip yang ingin Anda sertakan dalam pekerjaan yang satu ini. Anda dapat memilih parameter untuk setiap skrip. Anda juga dapat mengatur ulang skrip sehingga mereka berjalan dalam urutan yang benar.
2. Pada tab kedua, Anda dapat memberi nama pekerjaan Anda, dan langsung mengonfigurasi pengaturan untuk AWS Glue. Di layar ini, Anda dapat mengonfigurasi pengaturan berikut:
 - AWS Identity and Access Management (IAM) peran
 - Nama file skrip dan jalur file
 - Enkripsi skrip menggunakan enkripsi sisi server dengan kunci yang dikelola Amazon S3 (-S3 SSE)
 - Direktori sementara
 - Jalur pustaka Python yang dihasilkan
 - Jalur pustaka Python pengguna
 - Jalur untuk file.jar yang bergantung
 - Path file yang direferensikan
 - Bersamaan DPU untuk setiap pekerjaan yang dijalankan

- Konkurensi maksimum
 - Batas waktu kerja (dalam hitungan menit)
 - Ambang batas pemberitahuan tunda (dalam hitungan menit)
 - Jumlah percobaan
 - Konfigurasi keamanan
 - enkripsi di sisi server
3. Pada langkah ketiga, atau tab, Anda memilih koneksi yang dikonfigurasi ke titik akhir target.

Setelah Anda selesai mengkonfigurasi pekerjaan, itu akan ditampilkan di bawah ETL pekerjaan di Katalog AWS Glue Data. Jika Anda memilih pekerjaan, pengaturan akan ditampilkan sehingga Anda dapat meninjau atau mengeditnya. Untuk membuat pekerjaan baru AWS Glue, pilih Create AWS Glue Job dari menu konteks (klik kanan) untuk pekerjaan tersebut. Melakukan hal ini menerapkan definisi skema. Untuk menyegarkan tampilan, pilih Refresh dari database dari menu konteks (klik kanan).

Pada titik ini, Anda dapat melihat pekerjaan Anda di AWS Glue konsol. Untuk melakukannya, masuk ke AWS Management Console dan buka AWS Glue konsol di <https://console.aws.amazon.com/glue/>.

Anda dapat menguji pekerjaan baru untuk memastikan bahwa itu berfungsi dengan benar. Untuk melakukannya, pertama-tama periksa data di tabel sumber Anda, lalu verifikasi bahwa tabel target kosong. Jalankan pekerjaan, dan periksa lagi. Anda dapat melihat log kesalahan dari AWS Glue konsol.

Mengonversi ETL proses ke dalam AWS GlueAWS Schema Conversion Tool

Di bagian berikut, Anda dapat menemukan deskripsi konversi yang memanggil AWS Glue API operasi dengan Python. Untuk informasi selengkapnya, lihat [AWS Glue ETL Skrip program dengan Python di Panduan AWS Glue Pengembang](#).

Topik

- [Langkah 1: Buat database](#)
- [Langkah 2: Buat koneksi](#)
- [Langkah 3: Buat AWS Glue crawler](#)

Langkah 1: Buat database

Langkah pertama adalah membuat database baru dalam Katalog AWS Glue Data dengan menggunakan [AWS SDKAPI](#). Saat Anda menentukan tabel di Katalog Data, Anda menambahkannya ke database. Database digunakan untuk mengatur tabel di AWS Glue.

Contoh berikut menunjukkan `create_database` metode API AWS Glue Python untuk.

```
response = client.create_database(  
    DatabaseInput={  
        'Name': 'database_name',  
        'Description': 'description',  
        'LocationUri': 'string',  
        'Parameters': {  
            'parameter-name': 'parameter value'  
        }  
    }  
)
```

Jika Anda menggunakan Amazon Redshift, nama database dibentuk sebagai berikut.

```
{redshift_cluster_name}_{redshift_database_name}_{redshift_schema_name}
```

Nama lengkap cluster Amazon Redshift untuk contoh ini adalah sebagai berikut.

```
rsdbb03.apq1mpqso.us-west-2.redshift.amazonaws.com
```

Berikut ini menunjukkan contoh nama database yang terbentuk dengan baik. Dalam hal ini `rsdbb03` adalah nama, yang merupakan bagian pertama dari nama lengkap titik akhir cluster. Database diberi nama `dev` dan skema adalah `ora_glue`.

```
rsdbb03_dev_ora_glue
```

Langkah 2: Buat koneksi

Buat koneksi baru dalam Katalog Data dengan menggunakan [AWS SDKAPI](#).

Contoh berikut menunjukkan menggunakan [create_connection](#) metode API AWS Glue Python untuk.

```

response = client.create_connection(
    ConnectionInput={
        'Name': 'Redshift_abcde03.aabbcc112233.us-west-2.redshift.amazonaws.com_dev',
        'Description': 'Created from SCT',
        'ConnectionType': 'JDBC',
        'ConnectionProperties': {
            'JDBC_CONNECTION_URL': 'jdbc:redshift://aabbcc03.aabbcc112233.us-
west-2.redshift.amazonaws.com:5439/dev',
            'USERNAME': 'user_name',
            'PASSWORD': 'password'
        },
        'PhysicalConnectionRequirements': {
            'AvailabilityZone': 'us-west-2c',
            'SubnetId': 'subnet-a1b23c45',
            'SecurityGroupIdList': [
                'sg-000a2b3c', 'sg-1a230b4c', 'sg-aba12c3d', 'sg-1abb2345'
            ]
        }
    }
)

```

Parameter yang digunakan `create_connection` adalah sebagai berikut:

- **Name**(UTF-8 string) — diperlukan. Untuk Amazon Redshift, nama koneksi dibentuk sebagai berikut: `Redshift_<Endpoint-name>_<redshift-database-name>`, misalnya: `Redshift_abcde03_dev`
- **Description**(UTF-8 string) — Deskripsi Anda tentang koneksi.
- **ConnectionType**(UTF-8 string) - Diperlukan. Jenis koneksi. Saat ini, JDBC hanya didukung; SFTP tidak didukung.
- **ConnectionProperties**(dict) — Diperlukan. Daftar pasangan kunci-nilai yang digunakan sebagai parameter untuk koneksi ini, termasuk JDBC koneksiURL, nama pengguna, dan kata sandi.
- **PhysicalConnectionRequirements**(dict) — Persyaratan koneksi fisik, yang meliputi:
 - **SubnetId**(UTF-8 string) — ID subnet yang digunakan oleh koneksi.
 - **SecurityGroupIdList**(list) — Daftar ID grup keamanan yang digunakan oleh koneksi.
 - **AvailabilityZone**(UTF-8 string) - Diperlukan. Availability Zone yang berisi endpoint. Parameter ini tidak lagi digunakan.

Langkah 3: Buat AWS Glue crawler

Selanjutnya, Anda membuat AWS Glue crawler untuk mengisi katalog. AWS Glue Untuk informasi selengkapnya, lihat [Katalogisasi tabel dengan crawler di Panduan Pengembang](#). AWS Glue

Langkah pertama dalam menambahkan crawler adalah membuat database baru dalam Katalog Data dengan menggunakan [AWS SDKAPI](#). Sebelum Anda mulai, pastikan untuk terlebih dahulu menghapus versi sebelumnya dengan menggunakan `delete_crawler` operasi.

Saat Anda membuat crawler, beberapa pertimbangan berlaku:

- Untuk nama crawler, gunakan format `<redshift_node_name>_<redshift_database_name>_<redshift_shema_name>`, misalnya: `abcde03_dev_ora_glue`
- Gunakan IAM peran yang sudah ada. Untuk informasi selengkapnya tentang membuat IAM peran, lihat [Membuat IAM peran](#) di Panduan IAM Pengguna.
- Gunakan nama database yang Anda buat di langkah sebelumnya.
- Gunakan `ConnectionName` parameter, yang diperlukan.
- Untuk path parameter, gunakan jalur ke JDBC target, misalnya: `dev/ora_glue/%`

Contoh berikut menghapus crawler yang ada dan kemudian membuat yang baru dengan menggunakan Python untuk API 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'
)

```

Buat dan kemudian jalankan crawler yang menghubungkan ke satu atau beberapa penyimpanan data, menentukan struktur data, dan menulis tabel ke dalam Katalog Data. Anda dapat menjalankan crawler sesuai jadwal, seperti yang ditunjukkan berikut.

```

response = client.start_crawler(
    Name='string'
)

```

Contoh ini menggunakan Amazon Redshift sebagai target. Jenis data Amazon Redshift dipetakan ke tipe AWS Glue data dengan cara berikut setelah crawler berjalan.

Jenis data Amazon Redshift	AWS Glue tipe data
smallint	smallint
integer	int
bigint	bigint

desimal	desimal (18,0)
desimal (p, s)	desimal (p, s)
real	double
double precision	double
boolean	boolean
char	string
varchar	string
varchar (n)	string
date	tanggal
timestamp	timestamp
timestampz	timestamp

Mengonversi skrip Informatica ETL dengan AWS Schema Conversion Tool

Anda dapat menggunakan antarmuka baris AWS SCT perintah (CLI) untuk mengonversi ETL skrip Informatica Anda sehingga Anda dapat menggunakan skrip dengan basis data target baru Anda. Konversi ini mencakup tiga langkah utama. Pertama, AWS SCT mengkonversi SQL kode yang tertanam dalam objek Informatica Anda. Selanjutnya, AWS SCT mengubah nama objek database sesuai dengan aturan migrasi yang Anda tentukan dalam proyek Anda. Akhirnya, AWS SCT mengarahkan koneksi ETL skrip Informatica Anda ke database target baru.

Anda dapat mengonversi ETL skrip Informatica sebagai bagian dari proyek konversi AWS SCT database. Pastikan Anda menambahkan basis data sumber dan target ke proyek saat mengonversi skrip InformaticaETL.

Untuk mengonversi ETL skrip Informatica, pastikan Anda menggunakan AWS SCT versi 1.0.667 atau lebih tinggi. Juga, biasakan diri Anda dengan antarmuka baris perintah AWS SCT. Untuk informasi selengkapnya, lihat [CLIREferensi untuk AWS Schema Conversion Tool](#).

Untuk mengonversi skrip Informatica menggunakan ETL AWS SCT

1. Buat AWS SCT CLI skrip baru atau edit template skenario yang ada. Misalnya, Anda dapat mengunduh dan mengedit `InformaticConversionTemplate.scts` template. Untuk informasi selengkapnya, lihat [Mendapatkan CLI skenario](#).
2. Unduh JDBC driver yang diperlukan untuk basis data sumber dan target Anda. Tentukan lokasi driver ini menggunakan `SetGlobalSettings` perintah. Juga, tentukan folder tempat AWS SCT dapat menyimpan file log.

Contoh kode berikut menunjukkan cara menambahkan jalur ke SQL driver Oracle dan Postgre ke pengaturan. AWS SCT Setelah Anda menjalankan contoh kode ini, AWS SCT menyimpan file log di `C:\sct_log` folder. Juga, AWS SCT menyimpan file log konsol di `C:\Temp\oracle_postgresql` folder.

```
SetGlobalSettings
  -save: 'true'
  -settings: '{"oracle_driver_file": "C:\\drivers\\ojdbc8.jar",
    "postgresql_driver_file": "C:\\drivers\\postgresql-42.2.19.jar" }'
/

SetGlobalSettings
  -save: 'false'
  -settings: '{
    "log_folder": "C:\\sct_log",
    "console_log_folder": "C:\\Temp\\oracle_postgresql"}'
```

3. Buat AWS SCT proyek baru. Masukkan nama dan lokasi proyek Anda.

Contoh kode berikut membuat `oracle_postgresql` proyek dalam `C:\Temp` folder.

```
CreateProject
  -name: 'oracle_postgresql'
  -directory: 'C:\Temp'
/
```

4. Tambahkan informasi koneksi tentang basis data sumber dan target Anda.

Contoh kode berikut menambahkan SQL database Oracle dan Postgre sebagai sumber dan target untuk proyek Anda. AWS SCT


```
AddSource
  -password: 'source_password'
  -port: '1521'
  -vendor: 'ORACLE'
  -name: 'ORACLE'
  -host: 'source_address'
  -database: 'ORCL'
  -user: 'source_user'
/
AddTarget
  -database: 'postgresql'
  -password: 'target_password'
  -port: '5432'
  -vendor: 'POSTGRESQL'
  -name: 'POSTGRESQL'
  -host: 'target_address'
  -user: 'target_user'
/
```

Pada contoh sebelumnya, ganti *source_user* and *target_user* dengan nama pengguna database Anda. Selanjutnya, ganti *source_password* and *target_password* dengan kata sandi Anda. Untuk *source_address* and *target_address*, masukkan alamat IP server basis data sumber dan target Anda.

Untuk terhubung ke database Oracle versi 19 dan lebih tinggi, gunakan nama layanan Oracle dalam perintah. AddSource Untuk melakukannya, tambahkan `-connectionType` parameter dan atur nilainya ke `'basic_service_name'`. Kemudian, tambahkan `-servicename` parameter dan atur nilainya ke nama layanan Oracle Anda. Untuk informasi selengkapnya tentang AddSource perintah, lihat [AWS Schema Conversion Tool CLI Referensi](#).

5. Buat aturan AWS SCT pemetaan baru, yang mendefinisikan mesin database target untuk setiap skema basis data sumber. Untuk informasi selengkapnya, lihat [Memetakan tipe data di AWS Schema Conversion Tool](#).

Contoh kode berikut membuat aturan pemetaan yang mencakup semua skema database Oracle sumber dan mendefinisikan SQL Postgre sebagai target migrasi.

```
AddServerMapping
  -sourceTreePath: 'Servers.ORACLE'
  -targetTreePath: 'Servers.POSTGRESQL'
```

```
/
```

6. Tambahkan informasi koneksi tentang sumber Informatica dan file target XML Anda.

Contoh kode berikut menambahkan XML file Informatica dari C:\Informatica_source dan C:\Informatica_target folder.

```
AddSource
  -name: 'INFA_SOURCE'
  -vendor: 'INFORMATICA'
  -mappingsFolder: 'C:\Informatica_source'
/
AddTarget
  -name: 'INFA_TARGET'
  -vendor: 'INFORMATICA'
  -mappingsFolder: 'C:\Informatica_target'
/
```

7. Buat aturan pemetaan lain untuk menentukan file Informatica target untuk XML file Informatica sumber Anda. XML

Contoh kode berikut membuat aturan pemetaan yang mencakup sumber dan target XML file Informatica yang digunakan dalam contoh sebelumnya.

```
AddServerMapping
  -sourceTreePath: 'ETL.INFA_SOURCE'
  -targetTreePath: 'ETL.INFA_TARGET'
/
```

8. Tentukan koneksi server database yang sesuai dengan referensi nama koneksi Informatica.

Contoh kode berikut mengonfigurasi pengalihan ETL skrip Informatica Anda dari sumber Anda ke database target baru. Contoh ini juga mengkonfigurasi variabel koneksi.

```
ConfigureInformaticaConnectionsRedirect
  -treePath: 'ETL.INFA_SOURCE.Files'
  -connections: '{
    "ConnectionNames": [
      {
        "name": "Oracle_src",
        "newName": "postgres",
        "treePath": "Servers.ORACLE"
```

```

}
]
"ConnectionVariables": [
{
    "name": "$Source",
    "treePath": "Servers.ORACLE"
}
]
}'
/

```

9. Konversikan skema basis data sumber Anda dan skrip InformaticaETL.

Contoh kode berikut mengonversi semua skema database Oracle sumber Anda dan file Informatica Anda. XML

```

Convert
  -treePath: 'Servers.ORACLE.Schemas.%'
/
Convert
  -treePath: 'ETL.INFA_SOURCE.Files'
/

```

10. (Opsional) Simpan proyek konversi Anda dan laporan penilaian. Laporan ini mencakup item tindakan konversi dan rekomendasi tentang cara menangani masing-masing item.

Contoh kode berikut menyimpan proyek Anda dan menyimpan salinan laporan penilaian sebagai PDF file di C:\Temp folder.

```

SaveProject
/
SaveReportPDF
  -treePath: 'ETL.INFA_SOURCE.Files'
  -file: 'C:\Temp\Informatica.pdf'
/

```

11. Simpan file Informatica XML Anda yang telah dikonversi.

Contoh kode berikut menyimpan XML file yang dikonversi dalam C:\Temp folder. Anda menentukan folder ini pada langkah sebelumnya menggunakan AddTarget perintah.

```

SaveTargetInformaticaXML

```

```
-treePath: 'ETL.INFA_TARGET.Files'  
/
```

12. Simpan skrip Anda sebagai `.scts` file dan jalankan menggunakan `RunSCTBatch` perintah di file AWS SCT CLI. Untuk informasi selengkapnya, lihat [AWS SCT CLI modus skrip](#).

Contoh berikut menjalankan `Informatica.scts` script dalam `C:\Temp` folder. Anda dapat menggunakan contoh ini di Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\Temp\Informatica.scts"
```

Jika Anda mengedit skrip `Informatica` sumber Anda, maka jalankan ETL skrip lagi. AWS SCT CLI

Konversi ke SSIS dengan AWS Glue AWS SCT

Berikut ini, Anda dapat menemukan cara mengonversi paket Microsoft SQL Server Integration Services (SSIS) untuk AWS Glue menggunakan AWS SCT.

Untuk mengonversi SSIS paket Microsoft AWS Glue, pastikan Anda menggunakan AWS SCT versi 1.0.642 atau yang lebih tinggi. Anda juga perlu memiliki SSIS proyek dengan ETL paket — `.dtsx`, `.conmgr`, dan `.params` file di folder lokal.

Anda tidak memerlukan SSIS server yang diinstal. Proses konversi melewati SSIS file lokal.

Untuk mengonversi SSIS paket untuk AWS Glue menggunakan AWS SCT

1. Buat proyek baru di AWS SCT atau buka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called “Memulai dan Mengelola proyek”](#).
2. Pilih Tambahkan sumber dari menu untuk menambahkan SSIS paket sumber baru ke proyek Anda.
3. Pilih Layanan Integrasi SQL Server dan lengkapi yang berikut ini:
 - Nama koneksi — Masukkan nama untuk koneksi Anda. AWS SCT menampilkan nama ini di pohon metadata.
 - SSISfolder paket - Pilih jalur ke folder SSIS proyek Anda dengan paket.

AWS SCT membaca file proyek (file dengan ekstensi `.dtsx`, `.conmgr` atau `.params`) dari folder lokal dan menguraikannya. Kemudian mengaturnya ke dalam AWS SCT pohon kategori.

4. Pilih Tambahkan target dari menu untuk menambahkan platform target baru untuk mengonversi SSIS paket sumber Anda.
5. Pilih AWS Gluedan lengkapi yang berikut ini:
 - Nama koneksi — Masukkan nama untuk koneksi Anda. AWS SCT menampilkan nama ini di pohon metadata.
 - Salin dari AWS profil — Pilih profil yang akan digunakan.
 - AWS kunci akses — Masukkan kunci AWS akses Anda.
 - AWS kunci rahasia — Masukkan kunci AWS rahasia Anda.
 - Wilayah — Pilih Wilayah AWS yang ingin Anda gunakan dari daftar.
 - Folder bucket Amazon S3 — Masukkan jalur folder untuk bucket Amazon S3 yang akan Anda gunakan.

Anda dapat menggunakan AWS Glue target virtual. Dalam hal ini, Anda tidak perlu menentukan kredensi koneksi. Untuk informasi selengkapnya, lihat [the section called “Pemetaan target virtual”](#).

6. Buat aturan pemetaan baru yang mencakup SSIS paket sumber dan AWS Glue target Anda. Untuk informasi selengkapnya, lihat [the section called “Pemetaan tipe data baru”](#).
7. Pada menu Tampilan, pilih Tampilan utama.
8. Dalam tampilan SSIS pohon, buka menu konteks (klik kanan) untuk Manajer koneksi, lalu pilih Konfigurasi koneksi.
9. Konfigurasi manajer koneksi proyek.

Untuk mengkonfigurasi pemetaan SSIS koneksi untuk manajer koneksi, tentukan AWS Glue koneksi untuk manajer SSIS koneksi yang sesuai. Pastikan AWS Glue koneksi Anda sudah dibuat.

- a. Di bawah Koneksi, pilih Koneksi proyek.
 - b. Untuk koneksi katalog Glue, pilih AWS Glue koneksi yang sesuai.
10. Konfigurasi manajer koneksi paket:
 - a. Di bawah Koneksi, pilih paket Anda.
 - b. Untuk koneksi katalog Glue, pilih AWS Glue koneksi yang sesuai.
 - c. Ulangi tindakan ini untuk semua koneksi yang tersedia untuk paket Anda.

11. Pilih Terapkan.
12. Konversikan paket Anda. Dalam tampilan pohon sumber, temukan Paket. Buka menu konteks (klik kanan) untuk paket Anda, lalu pilih Konversi paket.
13. Simpan skrip yang dikonversi ke Amazon S3. Dalam tampilan pohon target, temukan skrip Package. Buka menu konteks (klik kanan) untuk skrip yang dikonversi, lalu pilih Simpan ke S3.
14. Konfigurasi AWS Glue pekerjaan Anda. Dalam tampilan pohon target, temukan skrip Package. Buka menu konteks (klik kanan) untuk skrip yang dikonversi, lalu pilih Konfigurasi AWS Glue pekerjaan.
15. Lengkapi tiga bagian konfigurasi:
 - a. Lengkapi bagian aliran data Desain:
 - Strategi eksekusi — Pilih bagaimana pekerjaan Anda akan menjalankan ETL skrip. Pilih SEQUENTIAL untuk menjalankan skrip dalam urutan yang ditentukan dalam wizard. Pilih PARALLEL untuk menjalankan skrip secara paralel, mengabaikan urutan yang ditentukan dalam wizard.
 - Skrip — Pilih nama skrip Anda yang dikonversi.
 - Pilih Berikutnya.
 - b. Lengkapi bagian Properti Job:
 - Nama — Masukkan nama AWS Glue pekerjaan Anda.
 - IAM Peran — Pilih IAM peran yang digunakan untuk otorisasi sumber daya yang digunakan untuk menjalankan pekerjaan dan mengakses penyimpanan data.
 - Nama file skrip — Masukkan nama skrip Anda yang telah dikonversi.
 - File skrip jalur S3 - Masukkan jalur Amazon S3 ke skrip yang dikonversi.
 - Enkripsi skrip menggunakan SSE -S3 - Pilih opsi ini untuk melindungi data menggunakan enkripsi sisi server dengan kunci enkripsi yang dikelola Amazon S3 (-S3). SSE
 - Direktori sementara — Masukkan jalur Amazon S3 ke direktori sementara untuk hasil perantara. AWS Glue dan transformasi AWS Glue bawaan menggunakan direktori ini untuk membaca atau menulis ke Amazon Redshift.
 - AWS SCT secara otomatis menghasilkan jalur untuk pustaka Python. Anda dapat meninjau jalur ini di jalur pustaka python yang dihasilkan. Anda tidak dapat mengedit jalur yang dibuat secara otomatis ini. Untuk menggunakan pustaka Python tambahan, masukkan jalur di jalur pustaka Python Pengguna.

- Jalur pustaka python pengguna - Masukkan jalur untuk pustaka Python pengguna tambahan. Pisahkan jalur Amazon S3 dengan koma.
- Jalur jar dependen - Masukkan jalur untuk file jar dependen. Pisahkan jalur Amazon S3 dengan koma.
- Jalur file yang direferensikan — Masukkan jalur untuk file tambahan, seperti file konfigurasi, yang diperlukan oleh skrip Anda. Pisahkan jalur Amazon S3 dengan koma.
- Kapasitas maksimum — Masukkan jumlah maksimum unit pemrosesan AWS Glue data (DPU) yang dapat dialokasikan saat pekerjaan ini berjalan. Anda dapat memasukkan bilangan bulat dari 2 hingga 100. Defaultnya adalah 2.
- Konkurensi maksimum - Masukkan jumlah maksimum proses bersamaan yang diizinkan untuk pekerjaan ini. Defaultnya adalah 1. AWS Glue mengembalikan kesalahan saat ambang batas ini tercapai.
- Job timeout (menit) — Masukkan nilai batas waktu pada ETL pekerjaan Anda sebagai perlindungan terhadap pekerjaan yang melarikan diri. Default-nya adalah 2880 menit (48 jam) untuk tugas batch. Jika pekerjaan melebihi batas ini, status job run berubah menjadi TIMEOUT.
- Ambang batas pemberitahuan tunda (menit) — Masukkan ambang batas dalam beberapa menit sebelum AWS SCT mengirim pemberitahuan penundaan.
- Jumlah percobaan ulang — Masukkan berapa kali (0-10) yang AWS Glue akan secara otomatis memulai ulang pekerjaan jika gagal. Pekerjaan yang mencapai batas waktu tunggu tidak dimulai ulang. Default-nya adalah 0.
- Pilih Berikutnya.

c. Konfigurasi koneksi yang diperlukan:

- i. Dari Semua koneksi, pilih AWS Glue koneksi yang diperlukan dan tambahkan ke daftar Koneksi yang dipilih.
- ii. Pilih Selesai.

16. Buat AWS Glue pekerjaan yang dikonfigurasi. Dalam tampilan pohon target, temukan dan perluas ETL pekerjaan. Buka menu konteks (klik kanan) untuk ETL pekerjaan yang Anda konfigurasi, lalu pilih Create AWS Glue Job.

17. Jalankan AWS Glue pekerjaan:

- a. Buka AWS Glue konsol di <https://console.aws.amazon.com/glue/>.
- b. Di panel navigasi, pilih Tugas.

- c. Pilih Tambah pekerjaan, lalu pilih pekerjaan yang ingin Anda jalankan.
- d. Pada tab Tindakan, pilih Jalankan pekerjaan.

SSISKomponen yang AWS SCT dapat dikonversi ke AWS Glue

Anda dapat menggunakan AWS SCT untuk mengonversi aliran data dan mengontrol komponen aliran, serta wadah, parameter, dan variabel.

Komponen aliran data yang didukung meliputi:

- ADONETDestinasi
- ADONETSumber
- Agregat
- Transformasi Cache
- Transformasi Peta Karakter
- Transformasi Split Bersyarat
- Salin Transformasi Kolom
- Transformasi Konversi Data
- Transformasi Kolom Berasal
- Tujuan Excel
- Sumber Excel
- Transformasi Kolom Ekspor
- Tujuan File Datar
- Sumber File Datar
- Transformasi Pencarian Fuzzy
- Transformasi Kolom Impor
- Transformasi Pencarian
- Gabung Gabung Transformasi
- Gabungkan Transformasi
- Transformasi Multicast
- ODBCTujuan
- ODBCSumber

- OLETransformasi Perintah DB
- OLETujuan DB
- OLESumber DB
- Transformasi Pengambilan Sampel Persentase
- Transformasi Pivot
- Tujuan File Mentah
- Sumber File Mentah
- RecordSet Destinasi
- Transformasi Hitungan Baris
- Transformasi Pengambilan Sampel Baris
- Urutkan Transformasi
- SQLTujuan Server
- Serikat Semua Transformasi
- Transformasi Unpivot
- XMLSumber

Komponen aliran kontrol yang didukung meliputi:

- Tugas Bulk Insert
- Tugas Execute Package
- Jalankan SQL Tugas
- Jalankan Tugas SQL Pernyataan T
- Tugas Ekspresi
- Tugas Sistem File
- Tugas Notify Operator
- Kirim Tugas Mail

SSISKontainer yang didukung meliputi:

- Untuk Kontainer Loop
- Wadah Foreach Loop
- Kontainer Urutan

Mengonversi SSIS paket ke dengan AWS Glue Studio

AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi paket Microsoft SQL Server Integration Services (SSIS) ke AWS Glue Studio.

SSIS Paket mencakup komponen yang diperlukan, seperti manajer koneksi, tugas, aliran kontrol, aliran data, parameter, event handler, dan variabel, untuk menjalankan tugas ekstrak, transformasi, dan load (ETL) tertentu. AWS SCT mengonversi SSIS paket ke format yang kompatibel dengan AWS Glue Studio. Setelah memigrasikan basis data sumber ke AWS Cloud, Anda dapat menjalankan AWS Glue Studio pekerjaan yang dikonversi ini untuk melakukan ETL tugas.

Untuk mengonversi SSIS paket Microsoft AWS Glue Studio, pastikan Anda menggunakan AWS SCT versi 1.0.661 atau yang lebih tinggi.

Topik

- [Prasyarat](#)
- [Menambahkan SSIS paket ke AWS SCT proyek Anda](#)
- [Mengonversi SSIS paket ke dengan AWS Glue Studio](#)
- [Membuat AWS Glue Studio pekerjaan menggunakan kode yang dikonversi](#)
- [Membuat laporan penilaian untuk SSIS paket dengan AWS SCT](#)
- [SSIS komponen yang AWS SCT dapat dikonversi ke AWS Glue Studio](#)

Prasyarat

Di bagian ini, pelajari tentang tugas prasyarat untuk konversi paket ke SSIS AWS Glue. Tugas-tugas ini termasuk membuat AWS sumber daya yang diperlukan di akun Anda.

Anda dapat menggunakan AWS Identity and Access Management (IAM) untuk menentukan kebijakan dan peran yang diperlukan untuk mengakses sumber daya yang AWS Glue Studio digunakan. Untuk informasi selengkapnya, lihat [IAM izin untuk AWS Glue Studio pengguna](#).

Setelah AWS SCT mengonversi skrip sumber Anda AWS Glue Studio, unggah skrip yang dikonversi ke bucket Amazon S3. Pastikan Anda membuat bucket Amazon S3 ini dan memilihnya di pengaturan profil AWS layanan. Untuk informasi selengkapnya tentang membuat bucket S3, lihat [Membuat bucket S3 pertama Anda](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Untuk memastikan bahwa AWS Glue Studio dapat terhubung ke penyimpanan data Anda, buat konektor khusus dan koneksi. Juga, simpan kredensi database di AWS Secrets Manager

Untuk membuat konektor khusus

1. Unduh JDBC driver untuk penyimpanan data Anda. Untuk informasi selengkapnya tentang JDBC driver yang AWS SCT menggunakan, lihat [Menginstal JDBC driver untuk AWS Schema Conversion Tool](#).
2. Unggah file driver ini ke bucket Amazon S3 Anda. Untuk informasi selengkapnya, lihat [Mengunggah objek ke bucket Anda](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
3. Masuk ke AWS Management Console dan buka AWS Glue Studio konsol di <https://console.aws.amazon.com/gluestudio/>.
4. Pilih Konektor, lalu pilih Buat konektor khusus.
5. Untuk Connector S3 URL, pilih Browse S3, dan pilih file JDBC driver yang Anda unggah ke bucket Amazon S3.
6. Masukkan Nama deskriptif untuk konektor Anda. Misalnya, masukkan **SQLServer**.
7. Untuk jenis Konektor, pilih JDBC.
8. Untuk nama Kelas, masukkan nama kelas utama untuk JDBC pengemudi Anda. Untuk SQL Server, masukkan **com.microsoft.sqlserver.jdbc.SQLServerDriver**.
9. Untuk JDBCURLbasis, masukkan JDBC pangkalanURL. Sintaks JDBC basis URL tergantung pada mesin database sumber Anda. Untuk SQL Server, gunakan format berikut: **jdbc:sqlserver://\$<host>:\$<port>;databaseName=\$<dbname>;user=\$<username>;password=\$<password>**.

Pastikan Anda mengganti *<host>*, *<port>*, *<dbname>*, *<username>*, dan *<password>* dengan nilai-nilai Anda.

10. Untuk pembatas URL parameter, masukkan titik koma (.) ;
11. Pilih Buat konektor.

Untuk menyimpan kredensi database di AWS Secrets Manager

1. Masuk ke AWS Management Console dan buka AWS Secrets Manager konsol di <https://console.aws.amazon.com/secretsmanager/>.
2. Pilih Simpan rahasia baru.

3. Pada halaman Pilih jenis rahasia, lakukan hal berikut:
 - a. Untuk tipe Rahasia, pilih jenis rahasia lainnya.
 - b. Untuk pasangan kunci/nilai, masukkan kunci berikut: **host**,,, **port dbnameusername**, dan **password**

Selanjutnya, masukkan nilai Anda untuk kunci-kunci ini.
4. Pada halaman Konfigurasi rahasia, masukkan nama Rahasia deskriptif. Misalnya, masukkan **SQL_Server_secret**.
5. Pilih Berikutnya. Kemudian, pada halaman Konfigurasi rotasi, pilih Berikutnya lagi.
6. Pada halaman Ulasan, tinjau detail rahasia Anda, lalu pilih Store.

Untuk membuat koneksi untuk konektor Anda

1. Masuk ke AWS Management Console dan buka AWS Glue Studio konsol di <https://console.aws.amazon.com/gluestudio/>.
2. Pilih konektor yang ingin Anda buat koneksi, lalu pilih Buat koneksi.
3. Pada halaman Buat koneksi, masukkan Nama deskriptif untuk koneksi Anda. Misalnya, masukkan **SQL-Server-connection**.
4. Untuk AWS Rahasia, pilih rahasia yang Anda buat AWS Secrets Manager.
5. Konfigurasi opsi Jaringan, lalu pilih Buat koneksi.

Sekarang, Anda dapat membuat AWS Glue Studio pekerjaan dengan konektor khusus. Untuk informasi selengkapnya, lihat [Menciptakan AWS Glue Studio pekerjaan](#).

Menambahkan SSIS paket ke AWS SCT proyek Anda

Anda dapat menambahkan beberapa SSIS paket ke satu AWS SCT proyek.

Untuk menambahkan SSIS paket ke AWS SCT proyek Anda

1. Buat proyek baru dengan AWS SCT atau buka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called "Memulai dan Mengelola proyek"](#).
2. Pilih Tambahkan sumber dari menu, lalu pilih Layanan Integrasi SQL Server.
3. Untuk nama Koneksi, masukkan nama untuk SSIS paket Anda. AWS SCT menampilkan nama ini di pohon di panel kiri.

4. Untuk folder SSIS paket, masukkan path ke folder dengan SSIS paket sumber.
5. Pilih Tambahkan target dari menu, lalu pilih AWS Glue Studio.

Untuk terhubung AWS Glue Studio, AWS SCT gunakan AWS profil Anda. Untuk informasi selengkapnya, lihat [Mengelola Profil di AWS Schema Conversion Tool](#).

6. Buat aturan pemetaan, yang mencakup SSIS paket sumber dan AWS Glue Studio target Anda. Untuk informasi selengkapnya, lihat [Memetakan tipe data di AWS Schema Conversion Tool](#).
7. Buat AWS Glue Studio koneksi di AWS Glue Studio konsol. Untuk informasi selengkapnya, lihat [Membuat koneksi untuk konektor](#).
8. Pilih Manajer koneksi di pohon kiri, buka menu konteks (klik kanan), lalu pilih Konfigurasi koneksi.

AWS SCT menampilkan jendela Konfigurasi koneksi.

9. Untuk setiap SSIS koneksi sumber, pilih AWS Glue Studio koneksi.

Mengonversi SSIS paket ke dengan AWS Glue StudioAWS SCT

Berikut, temukan cara mengonversi SSIS paket untuk AWS Glue Studio menggunakan AWS SCT.

Untuk mengonversi SSIS paket ke AWS Glue Studio

1. Tambahkan SSIS paket Anda ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Menambahkan SSIS paket ke AWS SCT proyek Anda](#).
2. Di panel kiri, perluas ETL dan SSIS node.
3. Pilih Paket, buka menu konteks (klik kanan), lalu pilih Konversi paket.

AWS SCT mengonversi SSIS paket yang Anda pilih menjadi JSON file. JSON objek-objek ini mewakili simpul dalam grafik asiklik terarah (). DAG Temukan file yang dikonversi di DAGs node Package di pohon kanan.

4. Pilih Package DAGs, buka menu konteks (klik kanan), lalu pilih Simpan ke Amazon S3.

Sekarang Anda dapat menggunakan skrip ini untuk membuat pekerjaan di AWS Glue Studio

Membuat AWS Glue Studio pekerjaan menggunakan kode yang dikonversi

Setelah mengonversi SSIS paket sumber, Anda dapat menggunakan JSON file yang dikonversi untuk membuat AWS Glue Studio pekerjaan.

Untuk membuat AWS Glue Studio pekerjaan

1. Pilih Package DAGs di pohon kanan, buka menu konteks (klik kanan), lalu pilih Configure AWS Glue Studio job.
2. (Opsional) Terapkan paket ekstensi yang mengemulasi SSIS fungsi di AWS Glue Studio.
3. Jendela Configure AWS Glue Studio job terbuka.

Lengkapi bagian Properti pekerjaan Dasar:

- Nama — Masukkan nama AWS Glue Studio pekerjaan Anda.
- Nama file skrip — Masukkan nama skrip pekerjaan Anda.
- Parameter Job — Tambahkan parameter dan masukkan nilainya.

Pilih Berikutnya.

4. Lengkapi bagian Properti pekerjaan lanjutan:
 - IAMPeran — Pilih IAM peran yang digunakan untuk otorisasi AWS Glue Studio dan akses penyimpanan data.
 - File skrip jalur S3 - Masukkan jalur Amazon S3 ke skrip yang dikonversi.
 - Direktori sementara — Masukkan jalur Amazon S3 ke direktori sementara untuk hasil perantara. AWS Glue Studio menggunakan direktori ini untuk membaca atau menulis ke Amazon Redshift.
 - AWS SCT secara otomatis menghasilkan jalur untuk pustaka Python. Anda dapat meninjau jalur ini di jalur pustaka python yang dihasilkan. Anda tidak dapat mengedit jalur yang dibuat secara otomatis ini. Untuk menggunakan pustaka Python tambahan, masukkan jalur di jalur pustaka Python Pengguna.
 - Jalur pustaka python pengguna - Masukkan jalur untuk pustaka Python pengguna tambahan. Pisahkan jalur Amazon S3 dengan koma.
 - Jalur jar dependen — Masukkan jalur untuk * . jar file dependen. Pisahkan jalur Amazon S3 dengan koma.

- Jalur file yang direferensikan — Masukkan jalur untuk file tambahan, seperti file konfigurasi, yang diperlukan oleh skrip Anda. Pisahkan jalur Amazon S3 dengan koma.
- Jenis pekerja — Pilih G.1X atau G.2X.

Ketika Anda memilih G.1X setiap pekerja memetakan ke 1 DPU (4 vCPU, 16 GB memori, dan 64 GB disk).

Ketika Anda memilih G.2X setiap pekerja memetakan ke 2 DPU (8 vCPU, 32 GB memori, dan 128 GB disk).

- Jumlah pekerja yang diminta — Masukkan jumlah pekerja yang dialokasikan saat pekerjaan berjalan.
- Konkurensi maksimum - Masukkan jumlah maksimum proses bersamaan yang diizinkan untuk pekerjaan ini. Defaultnya adalah 1. AWS Glue mengembalikan kesalahan saat ambang batas ini tercapai.
- Job timeout (menit) — Masukkan nilai batas waktu pada ETL pekerjaan Anda sebagai perlindungan terhadap pekerjaan yang melarikan diri. Standarnya adalah 2.880 menit (48 jam) untuk pekerjaan batch. Jika pekerjaan melebihi batas ini, status job run berubah menjadi TIMEOUT.
- Ambang batas pemberitahuan tunda (menit) — Masukkan ambang batas dalam beberapa menit sebelum AWS SCT mengirim pemberitahuan penundaan.
- Jumlah percobaan ulang — Masukkan berapa kali (0-10) yang AWS Glue akan secara otomatis memulai ulang pekerjaan jika gagal. Pekerjaan yang mencapai batas waktu tunggu tidak dimulai ulang. Default-nya adalah 0.

Pilih Selesai.

AWS SCT mengkonfigurasi pekerjaan yang Anda pilih AWS Glue Studio .

5. Temukan pekerjaan Anda yang dikonfigurasi di bawah ETL pekerjaan di pohon yang tepat. Pilih pekerjaan yang dikonfigurasi, buka menu konteks (klik kanan), lalu pilih Buat AWS Glue Studio pekerjaan.
6. Pilih status Lamar dan pastikan bahwa nilai Status untuk pekerjaan Anda adalah Sukses.
7. Buka AWS Glue Studio konsol, pilih Refresh, dan pilih pekerjaan Anda. Kemudian pilih Run.

Membuat laporan penilaian untuk SSIS paket dengan AWS SCT

Laporan penilaian ETL migrasi menyediakan informasi tentang mengonversi SSIS paket Anda ke format yang kompatibel dengannya AWS Glue Studio. Laporan penilaian mencakup item tindakan untuk komponen SSIS paket Anda. Item tindakan ini menunjukkan komponen mana yang tidak AWS SCT dapat dikonversi secara otomatis.

Untuk membuat laporan penilaian ETL migrasi

1. Perluas SSISnode di bawah ETLdi panel kiri.
2. Pilih Paket, buka menu konteks (klik kanan), lalu pilih Buat laporan.
3. Lihat tab Ringkasan. Di sini, AWS SCT menampilkan informasi ringkasan eksekutif dari laporan penilaian ETL migrasi. Ini termasuk hasil konversi untuk semua komponen SSIS paket Anda.
4. (Opsional) Simpan salinan lokal laporan penilaian ETL migrasi sebagai PDF file atau file values (CSV) yang dipisahkan koma:

- Untuk menyimpan laporan penilaian ETL migrasi sebagai PDF file, pilih Simpan ke PDF di kanan atas.

PDFfile tersebut berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.

- Untuk menyimpan laporan penilaian ETL migrasi sebagai CSV file, pilih Simpan ke CSV di kanan atas.

AWS SCT membuat tiga CSV file. File-file ini berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.

5. Pilih tab Item tindakan. Tab ini berisi daftar item yang memerlukan konversi manual ke AWS Glue Studio. Saat Anda memilih item tindakan dari daftar, AWS SCT sorot item dari SSIS paket sumber Anda tempat item tindakan berlaku.

SSISkomponen yang AWS SCT dapat dikonversi ke AWS Glue Studio

Anda dapat menggunakan AWS SCT untuk mengonversi komponen dan parameter aliran SSIS data ke AWS Glue Studio.

Komponen aliran data yang didukung meliputi:

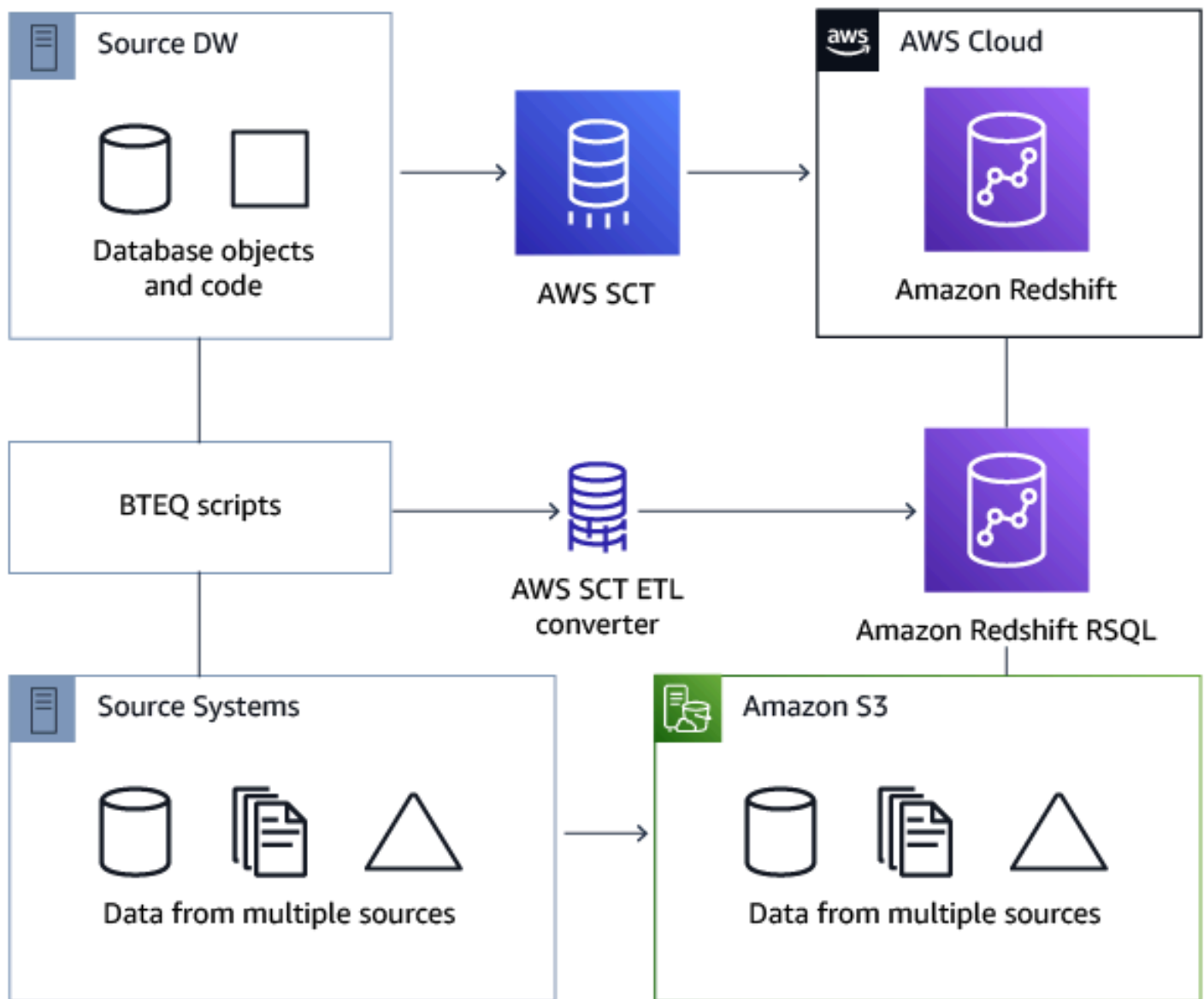
- ADONETDestinasi
- ADONETSumber
- Agregat
- Peta Karakter
- Split Bersyarat
- Salin Kolom
- Konversi Data
- Kolom Berasal
- Pencarian
- Gabungkan
- Gabung Bergabung
- Multicast
- ODBCDestination
- ODBCSource
- OLEDBDestination
- OLEDBSource
- Hitungan Baris
- Urutkan
- SQLTujuan Server
- Serikat Semua

AWS SCT dapat mengonversi lebih banyak SSIS komponen menjadi AWS Glue. Untuk informasi selengkapnya, lihat [SSISkomponen yang AWS SCT dapat dikonversi ke AWS Glue](#).

Mengonversi BTEQ skrip Teradata ke Amazon Redshift dengan RSQL AWS SCT

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skrip Teradata Basic Teradata Query (BTEQ) ke Amazon Redshift. RSQL

Diagram arsitektur berikut menunjukkan proyek migrasi database yang menyertakan konversi skrip ekstrak, transformasi, dan load (ETL) ke Amazon RSQL Redshift.



Topik

- [Menambahkan BTEQ skrip ke proyek Anda AWS SCT](#)
- [Mengkonfigurasi variabel substitusi dalam skrip dengan BTEQ AWS SCT](#)
- [Mengonversi BTEQ skrip Teradata ke Amazon Redshift dengan RSQL AWS SCT](#)
- [Mengelola BTEQ skrip dengan AWS SCT](#)
- [Membuat laporan penilaian konversi BTEQ skrip dengan AWS SCT](#)
- [Mengedit dan menyimpan BTEQ skrip Anda yang dikonversi dengan AWS SCT](#)

Menambahkan BTEQ skrip ke proyek Anda AWS SCT

Anda dapat menambahkan beberapa skrip ke satu AWS SCT proyek.

Untuk menambahkan BTEQ skrip ke AWS SCT proyek Anda

1. Buat proyek baru di AWS SCT atau buka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called “Memulai dan Mengelola proyek”](#).
2. Pilih Tambahkan sumber dari menu, lalu pilih Teradata untuk menambahkan basis data sumber Anda ke proyek. Untuk informasi selengkapnya, lihat [Database Teradata](#).
3. Pilih Tambahkan target dari menu untuk menambahkan database Amazon Redshift target ke proyek Anda AWS SCT .

Anda dapat menggunakan platform basis data target Amazon Redshift virtual. Untuk informasi selengkapnya, lihat [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).

4. Buat aturan pemetaan baru yang menyertakan database Teradata sumber Anda dan target Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Memetakan tipe data baru di AWS Schema Conversion Tool](#).
5. Pada menu Tampilan, pilih Tampilan utama.
6. Di panel kiri, perluas node Scripts.
7. Pilih BTEQskrip, buka menu konteks (klik kanan), lalu pilih Muat skrip.
8. Masukkan lokasi kode sumber untuk BTEQ skrip Teradata Anda dan pilih Pilih folder.

AWS SCT menampilkan jendela Load scripts.

9. Lakukan salah satu hal berikut ini:
 - a. Jika BTEQ skrip Teradata Anda tidak menyertakan variabel substitusi, pilih Tidak ada variabel substitusi, lalu pilih OK untuk menambahkan skrip ke proyek Anda. AWS SCT
 - b. Jika BTEQ skrip Teradata Anda menyertakan variabel substitusi, konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip BTEQ](#).

Mengkonfigurasi variabel substitusi dalam skrip dengan BTEQ AWS SCT

BTEQSkrip Teradata Anda dapat menyertakan variabel substitusi. Misalnya, Anda dapat menggunakan satu BTEQ skrip dengan variabel substitusi untuk menjalankan set perintah

yang sama pada beberapa lingkungan database. Anda dapat menggunakan AWS SCT untuk mengkonfigurasi variabel substitusi dalam BTEQ skrip Anda.

Sebelum Anda menjalankan BTEQ skrip dengan variabel substitusi, pastikan untuk menetapkan nilai untuk semua variabel. Untuk melakukan ini, Anda dapat menggunakan alat atau aplikasi lain seperti skrip Bash, UC4 (Automic), dan sebagainya. AWS SCT dapat menyelesaikan dan mengonversi variabel substitusi hanya setelah Anda menetapkan nilainya.

Untuk mengonfigurasi variabel substitusi dalam skrip Anda BTEQ

1. Tambahkan BTEQ skrip Anda ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Menambahkan BTEQ skrip ke proyek Anda AWS SCT](#).

Saat Anda menambahkan skrip Anda, pilih variabel Substitusi yang digunakan.

2. Untuk Tentukan format variabel, masukkan ekspresi reguler yang cocok dengan semua variabel substitusi dalam skrip Anda.

Misalnya, jika nama variabel substitusi Anda dimulai dengan `{` dan diakhiri dengan `}`, gunakan ekspresi `\${w+}` reguler. Untuk mencocokkan variabel substitusi yang dimulai dengan tanda dolar atau tanda persen, gunakan ekspresi `\${w+}|%\w+` reguler.

Ekspresi reguler AWS SCT sesuai dengan sintaks ekspresi reguler Java. Untuk informasi selengkapnya, lihat [java.util.regex Class](#) Pattern dalam dokumentasi Java.

3. Pilih OK untuk memuat skrip ke AWS SCT proyek Anda, lalu pilih OK untuk menutup jendela Load scripts.
4. Pilih Variabel untuk melihat semua variabel substitusi yang ditemukan dan nilainya.
5. Untuk Nilai, masukkan nilai untuk variabel substitusi.

Mengonversi BTEQ skrip Teradata ke Amazon Redshift dengan RSQL AWS SCT

Berikut ini, temukan cara mengonversi BTEQ ETL skrip ke Amazon RSQL AWS SCT Redshift menggunakan.

Untuk mengonversi BTEQ skrip Teradata ke Amazon Redshift RSQL

1. Tambahkan BTEQ skrip Anda ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Menambahkan BTEQ skrip ke proyek Anda AWS SCT](#).

2. Konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip BTEQ](#).
 3. Di panel kiri, perluas node Scripts.
 4. Lakukan salah satu hal berikut ini:
 - Untuk mengonversi BTEQ skrip tunggal, perluas node BTEQskrip, pilih skrip yang akan dikonversi, lalu pilih Konversi ke RSQL dari menu konteks (klik kanan).
 - Untuk menyembunyikan beberapa skrip, pastikan Anda memilih semua skrip untuk dikonversi. Kemudian pilih BTEQskrip, buka menu konteks (klik kanan), lalu pilih Konversi ke RSQL bawah Konversi skrip.
- AWS SCT mengonversi semua BTEQ skrip Teradata yang Anda pilih ke format yang kompatibel dengan Amazon Redshift. RSQL Temukan skrip Anda yang dikonversi di node Scripts di panel database target.
5. Edit RSQL skrip Amazon Redshift Anda yang telah dikonversi, atau simpan. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan BTEQ skrip Anda yang telah dikonversi](#).

Mengelola BTEQ skrip dengan AWS SCT

Anda dapat menambahkan beberapa BTEQ skrip atau menghapus BTEQ skrip dari AWS SCT proyek Anda.

Untuk menambahkan BTEQ skrip tambahan ke AWS SCT proyek Anda

1. Perluas node Scripts di panel kiri.
2. Pilih node BTEQskrip, dan buka menu konteks (klik kanan).
3. Pilih Muat skrip.
4. Masukkan informasi yang diperlukan untuk menambahkan BTEQ skrip baru dan mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, silakan lihat [Menambahkan BTEQ skrip ke proyek Anda AWS SCT](#) dan [Mengkonfigurasi variabel substitusi dalam skrip BTEQ](#).

Untuk menghapus BTEQ skrip dari AWS SCT proyek Anda

1. Perluas node BTEQscript di bawah Scripts di panel kiri.

2. Pilih skrip yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus skrip.

Membuat laporan penilaian konversi BTEQ skrip dengan AWS SCT

Laporan penilaian konversi BTEQ skrip memberikan informasi tentang mengonversi BTEQ perintah dan SQL pernyataan dari BTEQ skrip Anda ke format yang kompatibel dengan Amazon Redshift. RSQL Laporan penilaian mencakup item tindakan untuk BTEQ perintah dan SQL pernyataan yang tidak AWS SCT dapat dikonversi.

Untuk membuat laporan penilaian konversi BTEQ skrip

1. Perluas node BTEQscript di bawah Scripts di panel kiri.
2. Pilih skrip yang akan dikonversi, dan buka menu konteks (klik kanan).
3. Pilih Konversi ke RSQL bawah Buat laporan.
4. Lihat tab Ringkasan. Tab Ringkasan menampilkan informasi ringkasan eksekutif dari laporan penilaian BTEQ skrip. Ini termasuk hasil konversi untuk semua BTEQ perintah dan SQL pernyataan dari BTEQ skrip Anda.
5. (Opsional) Simpan salinan lokal laporan penilaian konversi BTEQ skrip sebagai PDF file atau file value (CSV) yang dipisahkan koma:
 - Untuk menyimpan laporan penilaian konversi BTEQ skrip sebagai PDF file, pilih Simpan ke PDF di kanan atas.

PDFFile tersebut berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.
 - Untuk menyimpan laporan penilaian konversi BTEQ skrip sebagai CSV file, pilih Simpan ke CSV di kanan atas.

CSVFile berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.
6. Pilih tab Item tindakan. Tab ini berisi daftar item yang memerlukan konversi manual ke Amazon RedshiftRSQL. Saat Anda memilih item tindakan dari daftar, AWS SCT sorot item dari BTEQ skrip sumber Anda yang berlaku untuk item tindakan.

Mengedit dan menyimpan BTEQ skrip Anda yang dikonversi dengan AWS SCT

Anda dapat mengedit skrip yang dikonversi di panel bawah AWS SCT proyek Anda. AWS SCT menyimpan skrip yang diedit sebagai bagian dari proyek Anda.

Untuk menyimpan skrip yang telah dikonversi

1. Perluas node RSQLscript di bawah Scripts di panel database target.
2. Pilih skrip yang dikonversi, buka menu konteks (klik kanan), dan pilih Simpan skrip.
3. Masukkan jalur ke folder untuk menyimpan skrip yang dikonversi dan pilih Simpan.

AWS SCT menyimpan skrip yang dikonversi ke file dan membuka file ini.

Mengonversi skrip shell dengan BTEQ perintah tertanam ke Amazon Redshift dengan RSQL AWS Schema Conversion Tool

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skrip shell dengan perintah Teradata Basic Teradata Query (BTEQ) tertanam ke skrip shell dengan perintah Amazon Redshift yang disematkan. RSQL

AWS SCT mengekstrak BTEQ perintah Teradata dari skrip shell Anda dan mengubahnya menjadi format yang kompatibel dengan Amazon Redshift. Setelah memigrasikan database Teradata ke Amazon Redshift, Anda dapat menggunakan skrip yang dikonversi ini untuk mengelola database Amazon Redshift baru Anda.

Anda juga dapat menggunakan AWS SCT untuk mengonversi file dengan BTEQ ETL skrip Teradata ke Amazon Redshift. RSQL Untuk informasi selengkapnya, lihat [Mengonversi BTEQ skrip Teradata ke Amazon Redshift dengan RSQL AWS SCT](#).

Topik

- [Menambahkan skrip shell dengan BTEQ perintah Teradata yang disematkan ke proyek Anda AWS SCT](#)
- [Mengkonfigurasi variabel substitusi dalam skrip shell dengan perintah Teradata yang disematkan BTEQ AWS SCT](#)
- [Mengonversi skrip shell dengan perintah BTEQ Teradata yang disematkan dengan AWS SCT](#)

- [Mengelola skrip shell dengan perintah BTEQ Teradata yang disematkan dengan AWS SCT](#)
- [Membuat laporan penilaian untuk konversi skrip shell dengan AWS SCT](#)
- [Mengedit dan menyimpan skrip shell Anda yang dikonversi dengan AWS SCT](#)

Menambahkan skrip shell dengan BTEQ perintah Teradata yang disematkan ke proyek Anda AWS SCT

Anda dapat menambahkan beberapa skrip ke satu AWS SCT proyek.

Untuk menambahkan skrip shell ke AWS SCT proyek Anda

1. Buat proyek baru di AWS SCT atau buka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called “Memulai dan Mengelola proyek”](#).
2. Pilih Tambahkan sumber dari menu, lalu pilih Teradata untuk menambahkan basis data sumber Anda ke proyek. Untuk informasi selengkapnya, lihat [Database Teradata](#).
3. Pilih Tambahkan target dari menu dan untuk menambahkan database Amazon Redshift target ke proyek Anda AWS SCT .

Anda dapat menggunakan platform basis data target Amazon Redshift virtual. Untuk informasi selengkapnya, lihat [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).

4. Buat aturan pemetaan baru yang menyertakan database Teradata sumber Anda dan target Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Memetakan tipe data baru di AWS Schema Conversion Tool](#).
5. Pada menu Tampilan, pilih Tampilan utama.
6. Di panel kiri, perluas node Scripts.
7. Pilih Shell, buka menu konteks (klik kanan), lalu pilih Muat skrip.
8. Masukkan lokasi skrip shell sumber Anda dengan BTEQ perintah Teradata tertanam dan pilih Pilih folder.

AWS SCT menampilkan jendela Load scripts.

9. Lakukan salah satu hal berikut ini:
 - Jika skrip shell Anda tidak menyertakan variabel substitusi, pilih Tidak ada variabel substitusi, lalu pilih OK untuk menambahkan skrip ke proyek Anda. AWS SCT

- Jika skrip shell Anda menyertakan variabel substitusi, konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip shell](#).

Mengkonfigurasi variabel substitusi dalam skrip shell dengan perintah Teradata yang disematkan BTEQ AWS SCT

Skrip shell Anda dapat menyertakan variabel substitusi. Misalnya, Anda dapat menggunakan skrip tunggal dengan variabel substitusi untuk mengelola database di lingkungan yang berbeda. Anda dapat menggunakan AWS SCT untuk mengkonfigurasi variabel substitusi dalam skrip shell Anda.

Sebelum Anda menjalankan BTEQ perintah dengan variabel substitusi dari skrip shell, pastikan untuk menetapkan nilai untuk semua variabel di dalam skrip shell ini. AWS SCT dapat menyelesaikan dan mengonversi variabel substitusi hanya setelah Anda menetapkan nilainya.

Untuk mengonfigurasi variabel substitusi dalam skrip shell Anda

1. Tambahkan skrip shell sumber Anda ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Menambahkan skrip shell ke proyek Anda AWS SCT](#).

Saat Anda menambahkan skrip Anda, pilih variabel Substitusi yang digunakan.

2. Untuk Tentukan format variabel, masukkan ekspresi reguler yang cocok dengan semua variabel substitusi dalam skrip Anda.

Misalnya, jika nama variabel substitusi Anda dimulai dengan `{` dan diakhiri dengan `}`, gunakan ekspresi `\${\w+}` reguler. Untuk mencocokkan variabel substitusi yang dimulai dengan tanda dolar atau tanda persen, gunakan ekspresi `\$\w+ |%\w+` reguler.

Ekspresi reguler AWS SCT sesuai dengan sintaks ekspresi reguler Java. Untuk informasi selengkapnya, lihat [java.util.regex Class](#) Pattern dalam dokumentasi Java.

3. Pilih OK untuk memuat skrip ke AWS SCT proyek Anda, lalu pilih OK untuk menutup jendela Load scripts.
4. Pilih Variabel untuk melihat semua variabel substitusi yang ditemukan dan nilainya.
5. Untuk Nilai, masukkan nilai untuk variabel substitusi.

Mengonversi skrip shell dengan perintah BTEQ Teradata yang disematkan dengan AWS SCT

Berikut ini, temukan cara mengonversi skrip shell dengan perintah Teradata yang disematkan ke skrip shell dengan BTEQ perintah Amazon Redshift yang disematkan menggunakan RSQL AWS SCT

Untuk mengonversi skrip shell

1. Tambahkan skrip shell Anda ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Menambahkan skrip shell ke proyek Anda AWS SCT](#).
2. Konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip shell](#).
3. Di panel kiri, perluas node Scripts.
4. Lakukan salah satu hal berikut ini:
 - Untuk mengonversi BTEQ perintah dari skrip shell tunggal, perluas simpul Shell, pilih skrip yang akan dikonversi, lalu pilih Konversi skrip dari menu konteks (klik kanan).
 - Untuk menyembunyikan beberapa skrip, pastikan Anda memilih semua skrip untuk dikonversi. Kemudian pilih Shell, buka menu konteks (klik kanan), lalu pilih Konversi skrip.
5. Pilih OKE.

AWS SCT mengonversi BTEQ perintah dalam skrip shell yang Anda pilih ke format yang kompatibel dengan Amazon Redshift. RSQL Temukan skrip Anda yang dikonversi di node Scripts di panel database target.

6. Edit RSQL skrip Amazon Redshift Anda yang telah dikonversi atau simpan. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan skrip shell Anda yang telah dikonversi](#).

Mengelola skrip shell dengan perintah BTEQ Teradata yang disematkan dengan AWS SCT

Anda dapat menambahkan beberapa skrip shell atau menghapus skrip shell dari AWS SCT proyek Anda.

Untuk menambahkan skrip shell baru ke AWS SCT proyek Anda

1. Perluas node Scripts di panel kiri.

2. Pilih simpul Shell, dan buka menu konteks (klik kanan).
3. Pilih Muat skrip.
4. Masukkan informasi yang diperlukan untuk menambahkan skrip shell baru dan mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, silakan lihat [Menambahkan skrip shell ke proyek Anda AWS SCT](#) dan [Mengkonfigurasi variabel substitusi dalam skrip shell](#).

Untuk menghapus skrip shell dari AWS SCT proyek Anda

1. Perluas node Shell di bawah Scripts di panel kiri.
2. Pilih skrip yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus skrip.

Membuat laporan penilaian untuk konversi skrip shell dengan AWS SCT

Laporan penilaian konversi skrip shell memberikan informasi tentang mengonversi BTEQ perintah dan SQL pernyataan. Konversi berasal dari skrip sumber Anda ke format yang kompatibel dengan Amazon RSQL Redshift. Laporan penilaian mencakup item tindakan untuk BTEQ perintah dan SQL pernyataan yang tidak AWS SCT dapat dikonversi.

Untuk membuat laporan penilaian konversi skrip shell

1. Perluas node Shell di bawah Scripts di panel kiri.
2. Pilih skrip yang akan dikonversi, buka menu konteks (klik kanan), lalu pilih Buat laporan.
3. Lihat tab Ringkasan. Tab Ringkasan menampilkan informasi ringkasan eksekutif dari laporan penilaian skrip shell. Ini mencakup hasil konversi untuk semua BTEQ perintah dan SQL pernyataan dari skrip sumber Anda.
4. (Opsional) Simpan salinan lokal laporan penilaian konversi skrip shell sebagai PDF file atau file values (CSV) yang dipisahkan koma:
 - Untuk menyimpan laporan penilaian konversi skrip shell sebagai PDF file, pilih Simpan ke PDF di kanan atas.

PDFFile tersebut berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.
 - Untuk menyimpan laporan penilaian konversi skrip shell sebagai CSV file, pilih Simpan ke CSV di kanan atas.

CSVFile berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.

5. Pilih tab Item tindakan. Tab ini berisi daftar item yang memerlukan konversi manual ke Amazon RedshiftRSQL. Saat Anda memilih item tindakan dari daftar, AWS SCT sorot item dari skrip shell sumber Anda tempat item tindakan berlaku.

Mengedit dan menyimpan skrip shell Anda yang dikonversi dengan AWS SCT

Anda dapat mengedit skrip yang dikonversi di panel bawah AWS SCT proyek Anda. AWS SCT menyimpan skrip yang diedit sebagai bagian dari proyek Anda.

Untuk menyimpan skrip yang telah dikonversi

1. Perluas node RSQLscript di bawah Scripts di panel database target.
2. Pilih skrip yang dikonversi, buka menu konteks (klik kanan), dan pilih Simpan skrip.
3. Masukkan jalur ke folder untuk menyimpan skrip yang dikonversi dan pilih Simpan.

AWS SCT menyimpan skrip yang dikonversi ke file dan membuka file ini.

Mengonversi FastExport skrip ke Amazon Redshift dengan RSQL AWS Schema Conversion Tool

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skrip FastExport pekerjaan Teradata ke Amazon Redshift. RSQL

Skrip FastExport pekerjaan adalah seperangkat FastExport perintah dan SQL pernyataan yang memilih dan mengekspor data dari database Teradata. AWS SCT mengonversi FastExport perintah dan SQL pernyataan ke format yang kompatibel dengan Amazon RSQL Redshift. Setelah memigrasikan database Teradata ke Amazon Redshift, Anda dapat menggunakan skrip yang dikonversi ini untuk mengekspor data dari database Amazon Redshift.

Topik

- [Menambahkan skrip FastExport pekerjaan ke proyek Anda AWS SCT](#)
- [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan Teradata FastExport dengan AWS SCT](#)

- [Mengonversi skrip pekerjaan Teradata FastExport dengan AWS SCT](#)
- [Mengelola skrip FastExport pekerjaan Teradata dengan AWS SCT](#)
- [Membuat laporan penilaian untuk konversi skrip FastExport pekerjaan Teradata dengan AWS SCT](#)
- [Mengedit dan menyimpan skrip FastExport pekerjaan Teradata Anda yang telah dikonversi dengan AWS SCT](#)

Menambahkan skrip FastExport pekerjaan ke proyek Anda AWS SCT

Anda dapat menambahkan beberapa skrip ke satu AWS SCT proyek.

Untuk menambahkan skrip FastExport pekerjaan ke AWS SCT proyek Anda

1. Buat proyek baru di AWS SCT atau buka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called “Memulai dan Mengelola proyek”](#).
2. Pilih Tambahkan sumber dari menu, lalu pilih Teradata untuk menambahkan basis data sumber Anda ke proyek. Untuk informasi selengkapnya, lihat [Database Teradata](#).
3. Pilih Tambahkan target dari menu dan untuk menambahkan database Amazon Redshift target ke proyek Anda AWS SCT .

Anda dapat menggunakan platform basis data target Amazon Redshift virtual. Untuk informasi selengkapnya, lihat [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).

4. Buat aturan pemetaan baru yang menyertakan database Teradata sumber Anda dan target Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Memetakan tipe data baru di AWS Schema Conversion Tool](#).
5. Pada menu Tampilan, pilih Tampilan utama.
6. Di panel kiri, perluas node Scripts.
7. Pilih FastExport, buka menu konteks (klik kanan), lalu pilih Muat skrip.
8. Masukkan lokasi kode sumber untuk skrip FastExport pekerjaan Teradata Anda dan pilih Pilih folder.

AWS SCT menampilkan jendela Load scripts.

9. Lakukan salah satu hal berikut ini:
 - Jika skrip FastExport pekerjaan Teradata Anda tidak menyertakan variabel substitusi, pilih Tidak ada variabel substitusi dan kemudian pilih OK untuk menambahkan skrip ke proyek Anda. AWS SCT

- Jika skrip FastExport pekerjaan Teradata Anda menyertakan variabel substitusi, konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan FastExport](#) .

Mengkonfigurasi variabel substitusi dalam skrip pekerjaan Teradata FastExport dengan AWS SCT

Skrip FastExport pekerjaan Teradata Anda dapat menyertakan variabel substitusi. Misalnya, Anda dapat menggunakan skrip tunggal dengan variabel substitusi untuk mengekspor data dari beberapa database. Anda dapat menggunakan AWS SCT untuk mengkonfigurasi variabel substitusi dalam skrip Teradata Anda.

Sebelum Anda menjalankan skrip FastExport pekerjaan dengan variabel substitusi, pastikan untuk menetapkan nilai untuk semua variabel. Untuk melakukan ini, Anda dapat menggunakan alat atau aplikasi lain seperti skrip Bash, UC4 (Automatic), dan sebagainya. AWS SCT dapat menyelesaikan dan mengonversi variabel substitusi hanya setelah Anda menetapkan nilainya.

Untuk mengonfigurasi variabel substitusi dalam skrip FastExport pekerjaan Anda

1. Tambahkan skrip FastExport pekerjaan Teradata sumber Anda ke proyek Anda. AWS SCT Untuk informasi selengkapnya, lihat [Menambahkan BTEQ skrip ke proyek Anda AWS SCT](#) .

Saat Anda menambahkan skrip Anda, pilih variabel Substitusi yang digunakan.

2. Untuk Tentukan format variabel, masukkan ekspresi reguler yang cocok dengan semua variabel substitusi dalam skrip Anda.

Misalnya, jika nama variabel substitusi Anda dimulai dengan `{` dan diakhiri dengan `}`, gunakan ekspresi `\\{\\w+\\}` reguler. Untuk mencocokkan variabel substitusi yang dimulai dengan tanda dolar atau tanda persen, gunakan ekspresi `\\$\\w+ | \\%\\w+` reguler.

Ekspresi reguler AWS SCT sesuai dengan sintaks ekspresi reguler Java. Untuk informasi selengkapnya, lihat [java.util.regex Class](#) Pattern dalam dokumentasi Java.

3. Pilih OK untuk memuat skrip ke AWS SCT proyek Anda, lalu pilih OK untuk menutup jendela Load scripts.
4. Di panel kiri, perluas node Scripts. Pilih FastExport, lalu pilih folder Anda dengan skrip. Buka menu konteks (klik kanan), lalu pilih Ekspor variabel di bawah Variabel substitusi.

5. Ekspor variabel substitusi untuk satu skrip. Perluas folder Anda dengan skrip, pilih skrip Anda, buka menu konteks (klik kanan), dan pilih Ekspor variabel di bawah Variabel substitusi.
6. Masukkan nama file values (CSV) dipisahkan koma untuk menyimpan variabel substitusi dan pilih Simpan.
7. Buka CSV file ini dan isi nilai untuk variabel substitusi.

Tergantung pada sistem operasi, AWS SCT menggunakan format yang berbeda untuk CSV file. Nilai dalam file mungkin terlampir dalam tanda kutip atau tidak. Pastikan Anda menggunakan format yang sama untuk nilai variabel substitusi seperti nilai lain dalam file. AWS SCT tidak dapat mengimpor CSV file dengan nilai dalam format yang berbeda.

8. Simpan file CSV.
9. Di panel kiri, perluas node Scripts. Pilih FastExport, lalu pilih skrip Anda. Buka menu konteks (klik kanan), lalu pilih Impor variabel di bawah Variabel substitusi.
10. Pilih CSV file Anda, lalu pilih Buka.
11. Pilih Variabel untuk melihat semua variabel substitusi yang ditemukan dan nilainya.

Mengonversi skrip pekerjaan Teradata FastExport dengan AWS SCT

Mengikuti, temukan cara mengonversi FastExport pekerjaan Teradata ke Amazon RSQL Redshift menggunakan. AWS SCT

Untuk mengonversi skrip FastExport pekerjaan Teradata ke Amazon Redshift RSQL

1. Tambahkan skrip FastExport pekerjaan Anda ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Menambahkan skrip FastExport pekerjaan ke proyek Anda AWS SCT](#).
2. Konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan FastExport](#).
3. Di panel kiri, perluas node Scripts.
4. Lakukan salah satu hal berikut ini:
 - Untuk mengonversi skrip FastExport pekerjaan tunggal, perluas FastExportsimpul, pilih skrip yang akan dikonversi, lalu pilih Konversi skrip dari menu konteks (klik kanan).
 - Untuk menyembunyikan beberapa skrip, pastikan Anda memilih semua skrip untuk dikonversi. Kemudian pilih FastExport, buka menu konteks (klik kanan), lalu pilih Konversi skrip.

AWS SCT mengonversi semua skrip FastExport pekerjaan Teradata yang Anda pilih ke format yang kompatibel dengan Amazon Redshift. RSQL Temukan skrip Anda yang dikonversi di node Scripts di panel database target.

5. Edit RSQL skrip Amazon Redshift Anda yang telah dikonversi atau simpan. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan skrip FastExport pekerjaan Anda yang telah dikonversi](#).

Mengelola skrip FastExport pekerjaan Teradata dengan AWS SCT

Anda dapat menambahkan beberapa skrip FastExport pekerjaan Teradata atau menghapus skrip FastExport pekerjaan dari proyek Anda. AWS SCT

Untuk menambahkan skrip FastExport pekerjaan baru ke AWS SCT proyek Anda

1. Perluas node Scripts di panel kiri.
2. Pilih FastExportnode, dan buka menu konteks (klik kanan).
3. Pilih Muat skrip.
4. Masukkan informasi yang diperlukan untuk menambahkan skrip FastExport pekerjaan baru dan mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, silakan lihat [Menambahkan skrip FastExport pekerjaan ke proyek Anda AWS SCT](#) dan [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan FastExport](#).

Untuk menghapus skrip FastExport pekerjaan dari AWS SCT proyek Anda

1. Perluas FastExportnode di bawah Skrip di panel kiri.
2. Pilih skrip yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus skrip.

Membuat laporan penilaian untuk konversi skrip FastExport pekerjaan Teradata dengan AWS SCT

Laporan penilaian konversi skrip FastExport pekerjaan memberikan informasi tentang mengonversi FastExport perintah dan SQL pernyataan dari FastExport skrip Anda ke format yang kompatibel

dengan Amazon Redshift. RSQL Laporan penilaian mencakup item tindakan untuk FastExport perintah dan SQL pernyataan yang tidak AWS SCT dapat dikonversi.

Untuk membuat laporan penilaian konversi skrip untuk pekerjaan Teradata FastExport

1. Perluas FastExportnode di bawah Skrip di panel kiri.
2. Pilih skrip yang akan dikonversi, buka menu konteks (klik kanan), lalu pilih Buat laporan.
3. Lihat tab Ringkasan. Tab Ringkasan menampilkan informasi ringkasan eksekutif dari laporan penilaian skrip FastExport pekerjaan. Ini mencakup hasil konversi untuk semua FastExport perintah dan SQL pernyataan dari skrip sumber Anda.
4. Anda dapat menyimpan salinan lokal laporan penilaian konversi skrip FastExport pekerjaan sebagai PDF file atau file values (CSV) yang dipisahkan koma.
 - a. Untuk menyimpan laporan penilaian konversi skrip FastExport pekerjaan sebagai PDF file, pilih Simpan ke PDF di kanan atas.

PDFFile tersebut berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.
 - b. Untuk menyimpan laporan penilaian konversi skrip FastExport pekerjaan sebagai CSV file, pilih Simpan ke CSV di kanan atas.

CSVFile berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.
5. Pilih tab Item tindakan. Tab ini berisi daftar item yang memerlukan konversi manual ke Amazon RedshiftRSQL. Saat Anda memilih item tindakan dari daftar, AWS SCT sorot item dari skrip FastExport pekerjaan sumber Anda yang berlaku untuk item tindakan tersebut.

Mengedit dan menyimpan skrip FastExport pekerjaan Teradata Anda yang telah dikonversi dengan AWS SCT

Anda dapat mengedit skrip yang dikonversi di panel bawah AWS SCT proyek Anda. AWS SCT menyimpan skrip yang diedit sebagai bagian dari proyek Anda.

Untuk menyimpan skrip yang telah dikonversi

1. Perluas node RSQLscript di bawah Scripts di panel database target.
2. Pilih skrip yang dikonversi, buka menu konteks (klik kanan), dan pilih Simpan skrip.

3. Masukkan jalur ke folder untuk menyimpan skrip yang dikonversi dan pilih Simpan.

AWS SCT menyimpan skrip yang dikonversi ke file dan membuka file ini.

Mengonversi skrip FastLoad pekerjaan ke Amazon Redshift dengan RSQL AWS Schema Conversion Tool

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi skrip FastLoad pekerjaan Teradata ke Amazon Redshift. RSQL

FastLoad Skrip Teradata adalah sekumpulan perintah yang menggunakan beberapa sesi untuk memuat data dalam tabel kosong pada Database Teradata. Teradata FastLoad memproses serangkaian perintah dan pernyataan Teradata FastLoad . SQL FastLoad Perintah Teradata menyediakan kontrol sesi dan penanganan data transfer data. SQLPernyataan membuat, memelihara, dan menjatuhkan tabel.

AWS SCT mengonversi FastLoad perintah dan SQL pernyataan Teradata ke format yang kompatibel dengan Amazon Redshift. RSQL Setelah memigrasikan database Teradata ke Amazon Redshift, Anda dapat menggunakan skrip yang dikonversi ini untuk memuat data ke database Amazon Redshift Anda.

Topik

- [Menambahkan skrip FastLoad pekerjaan ke proyek Anda AWS SCT](#)
- [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan Teradata FastLoad dengan AWS SCT](#)
- [Mengonversi skrip pekerjaan Teradata FastLoad dengan AWS SCT](#)
- [Mengelola skrip FastLoad pekerjaan Teradata dengan AWS SCT](#)
- [Membuat laporan penilaian untuk konversi skrip FastLoad pekerjaan Teradata dengan AWS SCT](#)
- [Mengedit dan menyimpan skrip FastLoad pekerjaan Teradata Anda yang telah dikonversi dengan AWS SCT](#)

Menambahkan skrip FastLoad pekerjaan ke proyek Anda AWS SCT

Anda dapat menambahkan beberapa skrip ke satu AWS SCT proyek.

Untuk menambahkan skrip FastLoad pekerjaan ke AWS SCT proyek Anda

1. Buat proyek baru di AWS SCT, atau buka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called “Memulai dan Mengelola proyek”](#).
2. Pilih Tambahkan sumber dari menu, lalu pilih Teradata untuk menambahkan basis data sumber Anda ke proyek. Untuk informasi selengkapnya, lihat [Database Teradata](#).
3. Pilih Tambahkan target dari menu dan tambahkan database Amazon Redshift target ke proyek Anda AWS SCT .

Anda dapat menggunakan platform basis data target Amazon Redshift virtual. Untuk informasi selengkapnya, lihat [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).

4. Buat aturan pemetaan baru yang menyertakan database Teradata sumber Anda dan target Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Memetakan tipe data baru di AWS Schema Conversion Tool](#).
5. Pada menu Tampilan, pilih Tampilan utama.
6. Di panel kiri, perluas node Scripts.
7. Pilih FastLoad, buka menu konteks (klik kanan), lalu pilih Muat skrip.
8. Masukkan lokasi skrip FastLoad pekerjaan Teradata sumber Anda dan pilih Pilih folder.

AWS SCT menampilkan jendela Load scripts.

9. Lakukan salah satu hal berikut ini:
 - Jika skrip FastLoad pekerjaan Teradata Anda tidak menyertakan variabel substitusi, pilih Tidak ada variabel substitusi, lalu pilih OK untuk menambahkan skrip ke proyek Anda. AWS SCT
 - Jika skrip FastLoad pekerjaan Teradata Anda menyertakan variabel substitusi, konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan FastLoad](#) .

Mengkonfigurasi variabel substitusi dalam skrip pekerjaan Teradata FastLoad dengan AWS SCT

Skrip FastLoad pekerjaan Teradata Anda mungkin menyertakan variabel substitusi. Misalnya, Anda dapat menggunakan skrip tunggal dengan variabel substitusi untuk memuat data ke database yang berbeda.

Sebelum Anda menjalankan skrip FastLoad pekerjaan dengan variabel substitusi, pastikan untuk menetapkan nilai untuk semua variabel. Untuk melakukan ini, Anda dapat menggunakan alat atau aplikasi lain seperti skrip Bash, UC4 (Automatic), dan sebagainya.

AWS SCT dapat menyelesaikan dan mengonversi variabel substitusi hanya setelah Anda menetapkan nilainya. Sebelum Anda memulai konversi skrip FastLoad pekerjaan Teradata sumber Anda, pastikan Anda menetapkan nilai untuk semua variabel substitusi. Anda dapat menggunakan AWS SCT untuk mengkonfigurasi variabel substitusi dalam skrip Teradata Anda.

Untuk mengonfigurasi variabel substitusi dalam skrip FastLoad pekerjaan Anda

1. Saat Anda menambahkan skrip FastLoad pekerjaan Teradata sumber ke AWS SCT proyek Anda, pilih Variabel substitusi yang digunakan. Untuk informasi selengkapnya tentang menambahkan skrip ini, lihat [Menambahkan skrip FastLoad pekerjaan ke proyek Anda AWS SCT](#).
2. Untuk Tentukan format variabel, masukkan ekspresi reguler yang cocok dengan semua variabel substitusi dalam skrip Anda.

Misalnya, jika nama variabel substitusi Anda dimulai dengan `{` dan diakhiri dengan `}`, gunakan ekspresi `\${\w+}` reguler. Untuk mencocokkan variabel substitusi yang dimulai dengan tanda dolar atau tanda persen, gunakan ekspresi `\$\w+ | \% \w+` reguler.

Ekspresi reguler AWS SCT sesuai dengan sintaks ekspresi reguler Java. Untuk informasi selengkapnya, lihat [java.util.regex Class](#) Pattern dalam dokumentasi Java.

3. Pilih OK untuk memuat skrip ke AWS SCT proyek Anda, lalu pilih OK untuk menutup jendela Load scripts.
4. Di panel kiri, perluas node Scripts. Pilih FastLoad, lalu pilih folder Anda dengan skrip. Buka menu konteks (klik kanan), lalu pilih Ekspor variabel di bawah Variabel substitusi.

Juga, Anda dapat mengekspor variabel substitusi untuk satu skrip. Perluas folder Anda dengan skrip, pilih skrip Anda, buka menu konteks (klik kanan), dan pilih Ekspor variabel di bawah Variabel substitusi.

5. Masukkan nama file value (CSV) dipisahkan koma untuk menyimpan variabel substitusi, lalu pilih Simpan.
6. Buka CSV file ini dan isi nilai untuk variabel substitusi.

Tergantung pada sistem operasi, AWS SCT menggunakan format yang berbeda untuk CSV file. Nilai dalam file mungkin terlampir dalam tanda kutip atau tidak. Pastikan Anda menggunakan

format yang sama untuk nilai variabel substitusi seperti nilai lain dalam file. AWS SCT tidak dapat mengimpor CSV file dengan nilai dalam format yang berbeda.

7. Simpan file CSV.
8. Di panel kiri, perluas node Scripts. Pilih FastLoad, lalu pilih skrip Anda. Buka menu konteks (klik kanan), lalu pilih Impor variabel di bawah Variabel substitusi.
9. Pilih CSV file Anda, lalu pilih Buka.
10. Pilih Variabel untuk melihat semua variabel substitusi yang ditemukan dan nilainya.

Mengonversi skrip pekerjaan Teradata FastLoad dengan AWS SCT

Mengikuti, temukan cara mengonversi FastLoad pekerjaan Teradata ke Amazon RSQL Redshift menggunakan AWS SCT

Untuk mengonversi skrip FastLoad pekerjaan Teradata ke Amazon Redshift RSQL

1. Tambahkan skrip FastLoad pekerjaan Anda ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Menambahkan skrip FastLoad pekerjaan ke proyek Anda AWS SCT](#).
2. Konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan FastLoad](#).
3. Di panel kiri, perluas node Scripts.
4. Lakukan salah satu hal berikut ini:
 - Untuk mengonversi skrip FastLoad pekerjaan tunggal, perluas FastLoadsimpul, pilih skrip yang akan dikonversi, lalu pilih Konversi skrip dari menu konteks (klik kanan).
 - Untuk menyembunyikan beberapa skrip, pastikan Anda memilih semua skrip untuk dikonversi. Pilih FastLoad, buka menu konteks (klik kanan), lalu pilih Konversi skrip. Kemudian lakukan salah satu hal berikut:
 - Jika Anda menyimpan file data sumber di Amazon S3, pilih jalur objek S3 untuk lokasi file data Sumber.

Masukkan nilai untuk folder bucket Amazon S3 dan bucket Amazon S3 untuk file manifes untuk file data sumber Anda.
 - Jika Anda tidak menyimpan file data sumber di Amazon S3, pilih Alamat host untuk lokasi file data Sumber.

Masukkan nilai untuk URL atau alamat IP host, nama login pengguna Host, dan bucket Amazon S3 untuk file manifest untuk file data sumber Anda.

5. Pilih OKE.

AWS SCT mengonversi semua skrip FastLoad pekerjaan Teradata yang Anda pilih ke format yang kompatibel dengan Amazon Redshift. RSQL Temukan skrip Anda yang dikonversi di node Scripts di panel database target.

6. Edit RSQL skrip Amazon Redshift Anda yang telah dikonversi atau simpan. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan skrip FastLoad pekerjaan Anda yang telah dikonversi](#).

Mengelola skrip FastLoad pekerjaan Teradata dengan AWS SCT

Anda dapat menambahkan beberapa skrip FastLoad pekerjaan Teradata atau menghapus skrip FastLoad pekerjaan dari proyek Anda. AWS SCT

Untuk menambahkan skrip FastLoad pekerjaan baru ke AWS SCT proyek Anda

1. Perluas node Scripts di panel kiri.
2. Pilih FastLoadnode dan buka menu konteks (klik kanan).
3. Pilih Muat skrip.
4. Masukkan informasi yang diperlukan untuk menambahkan skrip FastLoad pekerjaan baru dan mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, silakan lihat [Menambahkan skrip FastLoad pekerjaan ke proyek Anda AWS SCT](#) dan [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan FastLoad](#).

Untuk menghapus skrip FastLoad pekerjaan dari AWS SCT proyek Anda

1. Perluas FastLoadnode di bawah Skrip di panel kiri.
2. Pilih skrip yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus skrip.

Membuat laporan penilaian untuk konversi skrip FastLoad pekerjaan Teradata dengan AWS SCT

Laporan penilaian konversi skrip FastLoad pekerjaan memberikan informasi tentang mengonversi FastLoad perintah dan SQL pernyataan. Konversi berasal dari skrip sumber Anda ke format yang kompatibel dengan Amazon RSQL Redshift. Laporan penilaian mencakup item tindakan untuk FastLoad perintah dan SQL pernyataan yang tidak AWS SCT dapat dikonversi.

Untuk membuat laporan penilaian konversi skrip untuk pekerjaan Teradata FastLoad

1. Perluas FastLoadnode di bawah Skrip di panel kiri.
2. Pilih skrip yang akan dikonversi, buka menu konteks (klik kanan), lalu pilih Buat laporan.
3. Lihat tab Ringkasan.

Tab Ringkasan menampilkan informasi ringkasan eksekutif dari laporan penilaian skrip FastLoad pekerjaan. Ini mencakup hasil konversi untuk semua FastLoad perintah dan SQL pernyataan dari skrip sumber Anda.

4. (Opsional) Simpan salinan lokal laporan penilaian konversi skrip FastLoad pekerjaan sebagai PDF file atau file value (CSV) yang dipisahkan koma:
 - Untuk menyimpan laporan penilaian konversi skrip FastLoad pekerjaan sebagai PDF file, pilih Simpan ke PDF di kanan atas.

PDFFile tersebut berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.
 - Untuk menyimpan laporan penilaian konversi skrip FastLoad pekerjaan sebagai CSV file, pilih Simpan ke CSV di kanan atas.

CSVFile berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.
5. Pilih tab Item tindakan. Tab ini berisi daftar item yang memerlukan konversi manual ke Amazon RedshiftRSQL. Saat Anda memilih item tindakan dari daftar, AWS SCT sorot item dari skrip FastLoad pekerjaan sumber Anda yang berlaku untuk item tindakan tersebut.

Mengedit dan menyimpan skrip FastLoad pekerjaan Teradata Anda yang telah dikonversi dengan AWS SCT

Anda dapat mengedit skrip yang dikonversi di panel bawah AWS SCT proyek Anda. AWS SCT menyimpan skrip yang diedit sebagai bagian dari proyek Anda.

Untuk menyimpan skrip yang telah dikonversi

1. Perluas node RSQLscript di bawah Scripts di panel database target.
2. Pilih skrip yang dikonversi, buka menu konteks (klik kanan), dan pilih Simpan skrip.
3. Masukkan jalur ke folder untuk menyimpan skrip yang dikonversi dan pilih Simpan.

AWS SCT menyimpan skrip yang dikonversi ke file dan membuka file ini.

Mengonversi MultiLoad skrip ke Amazon Redshift dengan RSQL AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT untuk mengonversi skrip MultiLoad pekerjaan Teradata ke Amazon Redshift. RSQL

Skrip MultiLoad pekerjaan Teradata adalah seperangkat perintah untuk pemeliharaan batch Database Teradata Anda. Tugas MultiLoad impor Teradata melakukan sejumlah operasi penyisipan, pembaruan, dan penghapusan yang berbeda hingga lima tabel dan tampilan yang berbeda. Teradata MultiLoad menghapus tugas dapat menghapus sejumlah besar baris dari satu tabel.

AWS SCT mengonversi MultiLoad perintah dan SQL pernyataan Teradata ke format yang kompatibel dengan Amazon Redshift. RSQL Setelah memigrasikan database Teradata ke Amazon Redshift, gunakan skrip yang dikonversi ini untuk mengelola data di database Amazon Redshift Anda.

Topik

- [Menambahkan skrip MultiLoad pekerjaan ke proyek Anda AWS SCT](#)
- [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan Teradata MultiLoad dengan AWS SCT](#)
- [Mengonversi skrip pekerjaan Teradata MultiLoad dengan AWS SCT](#)
- [Mengelola skrip MultiLoad pekerjaan Teradata dengan AWS SCT](#)
- [Membuat laporan penilaian untuk konversi skrip MultiLoad pekerjaan Teradata dengan AWS SCT](#)

- [Mengedit dan menyimpan skrip MultiLoad pekerjaan Teradata Anda yang telah dikonversi dengan AWS SCT](#)

Menambahkan skrip MultiLoad pekerjaan ke proyek Anda AWS SCT

Anda dapat menambahkan beberapa skrip ke satu AWS SCT proyek.

Untuk menambahkan skrip MultiLoad pekerjaan ke AWS SCT proyek Anda

1. Buat proyek baru di AWS SCT atau buka proyek yang sudah ada. Untuk informasi selengkapnya, lihat [the section called “Memulai dan Mengelola proyek”](#).
2. Pilih Tambahkan sumber dari menu, lalu pilih Teradata untuk menambahkan basis data sumber Anda ke proyek. Untuk informasi selengkapnya, lihat [Database Teradata](#).
3. Pilih Tambahkan target dari menu dan untuk menambahkan database Amazon Redshift target ke proyek Anda AWS SCT .

Anda dapat menggunakan platform basis data target Amazon Redshift virtual. Untuk informasi selengkapnya, lihat [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).

4. Buat aturan pemetaan baru yang menyertakan database Teradata sumber Anda dan target Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Memetakan tipe data baru di AWS Schema Conversion Tool](#).
5. Pada menu Tampilan, pilih Tampilan utama.
6. Di panel kiri, perluas node Scripts.
7. Pilih MultiLoad, buka menu konteks (klik kanan), lalu pilih Muat skrip.
8. Masukkan lokasi skrip MultiLoad pekerjaan Teradata sumber Anda dan pilih Pilih folder.

AWS SCT menampilkan jendela Load scripts.

9. Lakukan salah satu hal berikut ini:
 - Jika skrip MultiLoad pekerjaan Teradata Anda tidak menyertakan variabel substitusi, pilih Tidak ada variabel substitusi, lalu pilih OK untuk menambahkan skrip ke proyek Anda. AWS SCT
 - Jika skrip MultiLoad pekerjaan Teradata Anda menyertakan variabel substitusi, konfigurasi variabel substitusi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan MultiLoad](#) .

Mengkonfigurasi variabel substitusi dalam skrip pekerjaan Teradata MultiLoad dengan AWS SCT

Skrip MultiLoad pekerjaan Teradata Anda mungkin menyertakan variabel substitusi. Misalnya, Anda dapat menggunakan skrip tunggal dengan variabel substitusi untuk memuat data ke database yang berbeda.

Sebelum Anda menjalankan skrip MultiLoad pekerjaan dengan variabel substitusi, pastikan untuk menetapkan nilai untuk semua variabel. Untuk melakukan ini, Anda dapat menggunakan alat atau aplikasi lain seperti skrip Bash, UC4 (Automic), dan sebagainya.

AWS SCT dapat menyelesaikan dan mengonversi variabel substitusi hanya setelah Anda menetapkan nilainya. Sebelum Anda memulai konversi skrip MultiLoad pekerjaan Teradata sumber Anda, pastikan bahwa Anda menetapkan nilai untuk semua variabel substitusi. Anda dapat menggunakan AWS SCT untuk mengkonfigurasi variabel substitusi dalam skrip Teradata Anda.

Untuk mengonfigurasi variabel substitusi dalam skrip MultiLoad pekerjaan Anda

1. Saat Anda menambahkan skrip MultiLoad pekerjaan Teradata sumber ke AWS SCT proyek Anda, pilih Variabel substitusi yang digunakan. Untuk informasi selengkapnya tentang menambahkan skrip ini, lihat [Menambahkan skrip MultiLoad pekerjaan ke proyek Anda AWS SCT](#).
2. Untuk Tentukan format variabel, masukkan ekspresi reguler yang cocok dengan semua variabel substitusi dalam skrip Anda.

Misalnya, jika nama variabel substitusi Anda dimulai dengan `{` dan diakhiri dengan `}`, gunakan ekspresi `\${\w+}` reguler. Untuk mencocokkan variabel substitusi yang dimulai dengan tanda dolar atau tanda persen, gunakan ekspresi `\$\w+|\%\w+` reguler.

Ekspresi reguler AWS SCT sesuai dengan sintaks ekspresi reguler Java. Untuk informasi selengkapnya, lihat [java.util.regex Class](#) Pattern dalam dokumentasi Java.

3. Pilih OK untuk memuat skrip ke AWS SCT proyek Anda, lalu pilih OK untuk menutup jendela Load scripts.
4. Pilih Variabel untuk melihat semua variabel substitusi yang ditemukan dan nilainya.
5. Untuk Nilai, masukkan nilai untuk variabel substitusi.

Mengonversi skrip pekerjaan Teradata MultiLoad dengan AWS SCT

Mengikuti, temukan cara mengonversi MultiLoad pekerjaan Teradata ke Amazon RSQL Redshift menggunakan AWS SCT

Untuk mengonversi skrip MultiLoad pekerjaan Teradata ke Amazon Redshift RSQL

1. Tambahkan skrip MultiLoad pekerjaan Anda ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Menambahkan skrip MultiLoad pekerjaan ke proyek Anda AWS SCT](#).
2. Konfigurasi variabel substitusi dan masukkan nilainya. Untuk informasi selengkapnya, lihat [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan MultiLoad](#).
3. Di panel kiri, perluas node Scripts.
4. Lakukan salah satu hal berikut ini:
 - Untuk mengonversi skrip MultiLoad pekerjaan tunggal, perluas MultiLoadsimpul, pilih skrip yang akan dikonversi, lalu pilih Konversi skrip dari menu konteks (klik kanan).
 - Untuk menyembunyikan beberapa skrip, pastikan Anda memilih semua skrip untuk dikonversi. Pilih MultiLoad, buka menu konteks (klik kanan), lalu pilih Konversi skrip.
5. Lakukan salah satu hal berikut ini:
 - Jika Anda menyimpan file data sumber di Amazon S3, pilih jalur objek S3 untuk lokasi file data Sumber.

Masukkan folder bucket Amazon S3 dan bucket Amazon S3 untuk file manifes untuk file data sumber Anda.
 - Jika Anda tidak menyimpan file data sumber di Amazon S3, pilih Alamat host untuk lokasi file data Sumber.

Masukkan URL atau alamat IP host, nama login pengguna Host, dan bucket Amazon S3 untuk file manifes untuk file data sumber Anda.
6. Pilih OKE.

AWS SCT mengonversi semua skrip MultiLoad pekerjaan Teradata yang Anda pilih ke format yang kompatibel dengan Amazon Redshift. RSQL Temukan skrip Anda yang dikonversi di node Scripts di panel database target.

7. Edit RSQL skrip Amazon Redshift Anda yang telah dikonversi atau simpan. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan skrip MultiLoad pekerjaan Anda yang telah dikonversi](#).

Mengelola skrip MultiLoad pekerjaan Teradata dengan AWS SCT

Anda dapat menambahkan beberapa skrip MultiLoad pekerjaan Teradata atau menghapus skrip MultiLoad pekerjaan dari proyek Anda. AWS SCT

Untuk menambahkan skrip MultiLoad pekerjaan baru ke AWS SCT proyek Anda

1. Perluas node Scripts di panel kiri.
2. Pilih MultiLoadnode dan buka menu konteks (klik kanan).
3. Pilih Muat skrip.
4. Masukkan informasi yang diperlukan untuk menambahkan skrip MultiLoad pekerjaan baru dan mengkonfigurasi variabel substitusi. Untuk informasi selengkapnya, silakan lihat [Menambahkan skrip MultiLoad pekerjaan ke proyek Anda AWS SCT](#) dan [Mengkonfigurasi variabel substitusi dalam skrip pekerjaan MultiLoad](#).

Untuk menghapus skrip MultiLoad pekerjaan dari AWS SCT proyek Anda

1. Perluas MultiLoadnode di bawah Skrip di panel kiri.
2. Pilih skrip yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus skrip.

Membuat laporan penilaian untuk konversi skrip MultiLoad pekerjaan Teradata dengan AWS SCT

Laporan penilaian konversi skrip MultiLoad pekerjaan memberikan informasi tentang mengonversi MultiLoad perintah dan SQL pernyataan. Konversi berasal dari skrip sumber Anda ke perintah SQL dan pernyataan Amazon RSQL Redshift untuk Amazon Redshift. Laporan penilaian mencakup item tindakan untuk MultiLoad perintah dan SQL pernyataan yang tidak AWS SCT dapat dikonversi.

Untuk membuat laporan penilaian konversi skrip untuk pekerjaan Teradata MultiLoad

1. Perluas MultiLoadnode di bawah Skrip di panel kiri.

2. Pilih skrip untuk membuat laporan penilaian, buka menu konteks (klik kanan), lalu pilih Buat laporan.
3. Lihat tab Ringkasan. Tab Ringkasan menampilkan informasi ringkasan eksekutif dari laporan penilaian skrip MultiLoad pekerjaan. Ini mencakup hasil konversi untuk semua MultiLoad perintah dan SQL pernyataan dari skrip sumber Anda.
4. (Opsional) Simpan salinan lokal laporan penilaian konversi skrip MultiLoad pekerjaan sebagai PDF file atau file value (CSV) yang dipisahkan koma:
 - Untuk menyimpan laporan penilaian konversi skrip MultiLoad pekerjaan sebagai PDF file, pilih Simpan ke PDF di kanan atas.

PDFFile tersebut berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi skrip.
 - Untuk menyimpan laporan penilaian konversi skrip MultiLoad pekerjaan sebagai CSV file, pilih Simpan ke CSV di kanan atas.

AWS SCT membuat dua CSV file. File-file ini berisi ringkasan eksekutif, item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi skrip.
5. Pilih tab Item tindakan. Tab ini berisi daftar item yang memerlukan konversi manual ke Amazon RedshiftRSQL. Saat Anda memilih item tindakan dari daftar, AWS SCT sorot item dari skrip MultiLoad pekerjaan sumber Anda yang berlaku untuk item tindakan tersebut.

Mengedit dan menyimpan skrip MultiLoad pekerjaan Teradata Anda yang telah dikonversi dengan AWS SCT

Anda dapat mengedit skrip yang dikonversi di panel bawah AWS SCT proyek Anda. AWS SCT menyimpan skrip yang diedit sebagai bagian dari proyek Anda.

Untuk menyimpan skrip yang telah dikonversi

1. Perluas node RSQLscript di bawah Scripts di panel database target.
2. Pilih skrip yang dikonversi, buka menu konteks (klik kanan), dan pilih Simpan skrip.
3. Masukkan jalur ke folder untuk menyimpan skrip yang dikonversi dan pilih Simpan.

AWS SCT menyimpan skrip yang dikonversi ke file dan membuka file ini.

Migrasi kerangka kerja data besar dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk memigrasikan kerangka kerja data besar ke file. AWS Cloud

Saat ini, AWS SCT mendukung migrasi cluster Hadoop ke Amazon dan Amazon EMR S3. Proses migrasi ini mencakup Hive dan HDFS layanan.

Selain itu, Anda dapat menggunakan AWS SCT untuk mengotomatiskan konversi alur kerja orkestrasi Apache Oozie Anda ke. AWS Step Functions

Topik

- [Memigrasi beban kerja Hadoop ke Amazon dengan EMR AWS Schema Conversion Tool](#)
- [Mengonversi workflowas Oozie menjadi dengan AWS Step FunctionsAWS Schema Conversion Tool](#)

Memigrasi beban kerja Hadoop ke Amazon dengan EMR AWS Schema Conversion Tool

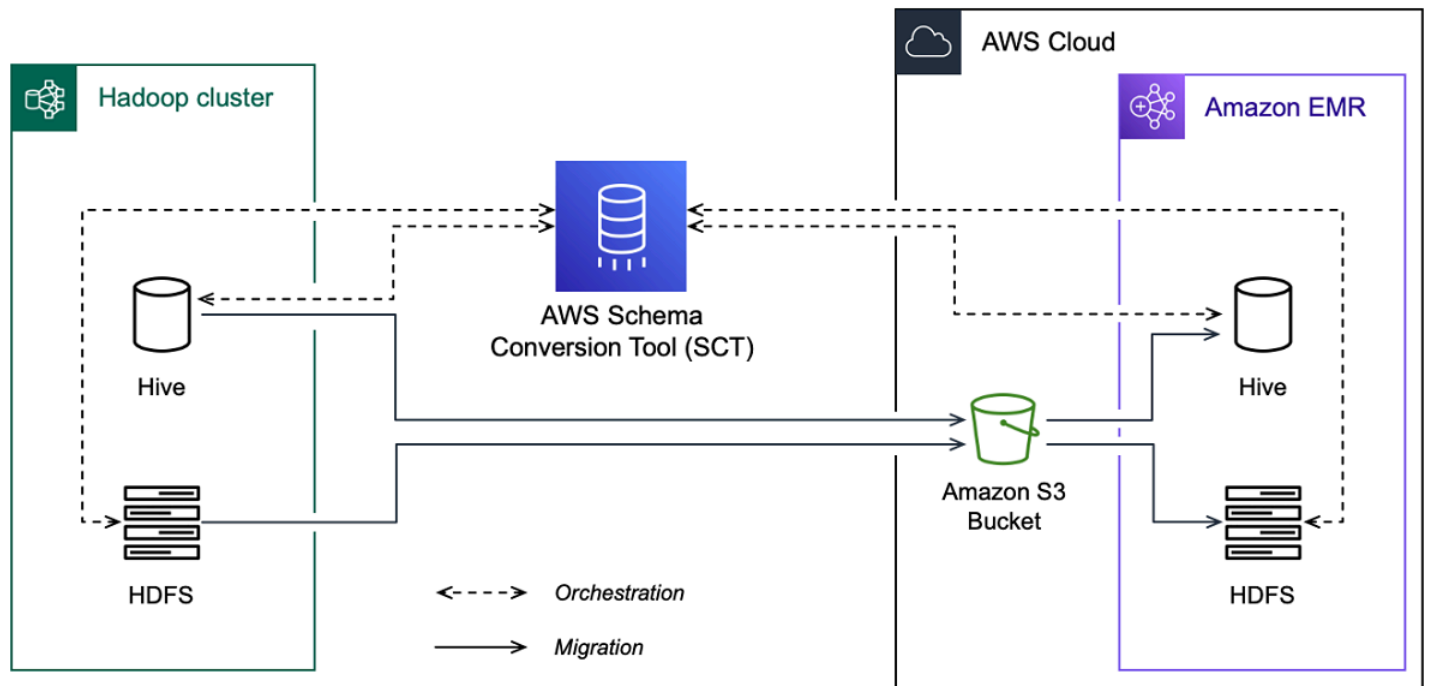
Untuk memigrasikan cluster Apache Hadoop, pastikan Anda menggunakan AWS SCT versi 1.0.670 atau yang lebih tinggi. Juga, biasakan diri Anda dengan antarmuka baris perintah (CLI) dari AWS SCT. Untuk informasi selengkapnya, lihat [CLIREferensi untuk AWS Schema Conversion Tool](#).

Topik

- [Ikhtisar migrasi](#)
- [Langkah 1: Connect ke cluster Hadoop Anda](#)
- [Langkah 2: Siapkan aturan pemetaan](#)
- [Langkah 3: Buat laporan penilaian](#)
- [Langkah 4: Migrasikan cluster Apache Hadoop Anda ke Amazon dengan EMR AWS SCT](#)
- [Menjalankan CLI skrip Anda](#)
- [Mengelola proyek migrasi data besar Anda](#)

Ikhtisar migrasi

Gambar berikut menunjukkan diagram arsitektur migrasi dari Apache Hadoop ke Amazon. EMR



AWS SCT memigrasikan data dan metadata dari cluster Hadoop sumber Anda ke bucket Amazon S3. Selanjutnya, AWS SCT gunakan metadata Hive sumber Anda untuk membuat objek database di layanan Amazon EMR Hive target. Secara opsional, Anda dapat mengkonfigurasi Hive untuk menggunakan AWS Glue Data Catalog sebagai metastore nya. Dalam hal ini, AWS SCT memigrasikan metadata Hive sumber Anda ke file. AWS Glue Data Catalog

Kemudian, Anda dapat menggunakannya AWS SCT untuk memigrasikan data dari bucket Amazon S3 ke layanan Amazon EMR HDFS target Anda. Atau, Anda dapat meninggalkan data di bucket Amazon S3 dan menggunakannya sebagai repositori data untuk beban kerja Hadoop Anda.

Untuk memulai migrasi Hadoop, Anda membuat dan menjalankan AWS SCT CLI skrip Anda. Skrip ini mencakup set lengkap perintah untuk menjalankan migrasi. Anda dapat mengunduh dan mengedit templat skrip migrasi Hadoop. Untuk informasi selengkapnya, lihat [Mendapatkan CLI skenario](#).

Pastikan skrip Anda menyertakan langkah-langkah berikut sehingga Anda dapat menjalankan migrasi dari Apache Hadoop ke Amazon S3 dan Amazon. EMR

Langkah 1: Connect ke cluster Hadoop Anda

Untuk memulai migrasi cluster Apache Hadoop Anda, buat proyek baru. AWS SCT Selanjutnya, sambungkan ke cluster sumber dan target Anda. Pastikan Anda membuat dan menyediakan AWS sumber daya target sebelum memulai migrasi.

Pada langkah ini, Anda menggunakan AWS SCT CLI perintah berikut.

- `CreateProject`— untuk membuat AWS SCT proyek baru.
- `AddSourceCluster`— untuk terhubung ke cluster Hadoop sumber di proyek Anda AWS SCT .
- `AddSourceClusterHive`— untuk terhubung ke layanan sumber Hive di proyek Anda.
- `AddSourceClusterHDFS`— untuk terhubung ke HDFS layanan sumber dalam proyek Anda.
- `AddTargetCluster`— untuk terhubung ke EMR kluster Amazon target di proyek Anda.
- `AddTargetClusterS3`— untuk menambahkan bucket Amazon S3 ke proyek Anda.
- `AddTargetClusterHive`— untuk terhubung ke layanan Hive target di proyek Anda
- `AddTargetClusterHDFS`— untuk terhubung ke HDFS layanan target dalam proyek Anda

Untuk contoh menggunakan AWS SCT CLI perintah ini, lihat [Menghubungkan ke Apache Hadoop](#).

Ketika Anda menjalankan perintah yang menghubungkan ke sumber atau target cluster, AWS SCT mencoba untuk membuat koneksi ke cluster ini. Jika upaya koneksi gagal, maka AWS SCT berhenti menjalankan perintah dari CLI skrip Anda dan menampilkan pesan kesalahan.

Langkah 2: Siapkan aturan pemetaan

Setelah Anda terhubung ke sumber dan kluster target, atur aturan pemetaan. Aturan pemetaan mendefinisikan target migrasi untuk kluster sumber. Pastikan Anda menyiapkan aturan pemetaan untuk semua kluster sumber yang ditambahkan dalam proyek Anda AWS SCT . Untuk informasi selengkapnya tentang aturan pemetaan, lihat [Memetakan tipe data di AWS Schema Conversion Tool](#).

Pada langkah ini, Anda menggunakan `AddServerMapping` perintah. Perintah ini menggunakan dua parameter, yang menentukan cluster sumber dan target. Anda dapat menggunakan `AddServerMapping` perintah dengan jalur eksplisit ke objek database Anda atau dengan nama objek. Untuk opsi pertama, Anda menyertakan jenis objek dan namanya. Untuk opsi kedua, Anda hanya menyertakan nama objek.

- `sourceTreePath`— jalur eksplisit ke objek database sumber Anda.

`targetTreePath`— jalur eksplisit ke objek database target Anda.

- `sourceNamePath`— jalur yang hanya mencakup nama-nama objek sumber Anda.

`targetNamePath`— jalur yang hanya mencakup nama-nama objek target Anda.

Contoh kode berikut membuat aturan pemetaan menggunakan jalur eksplisit untuk database sumber `testdb` Hive dan cluster target. EMR

```
AddServerMapping
  -sourceTreePath: 'Clusters.HADOOP_SOURCE.HIVE_SOURCE.Databases.testdb'
  -targetTreePath: 'Clusters.HADOOP_TARGET.HIVE_TARGET'
/
```

Anda dapat menggunakan contoh ini dan contoh berikut di Windows. Untuk menjalankan CLI perintah di Linux, pastikan Anda memperbarui jalur file dengan tepat untuk sistem operasi Anda.

Contoh kode berikut membuat aturan pemetaan menggunakan jalur yang hanya menyertakan nama objek.

```
AddServerMapping
  -sourceNamePath: 'HADOOP_SOURCE.HIVE_SOURCE.testdb'
  -targetNamePath: 'HADOOP_TARGET.HIVE_TARGET'
/
```

Anda dapat memilih Amazon EMR atau Amazon S3 sebagai target untuk objek sumber Anda. Untuk setiap objek sumber, Anda dapat memilih hanya satu target dalam satu AWS SCT proyek. Untuk mengubah target migrasi objek sumber, hapus aturan pemetaan yang ada, lalu buat aturan pemetaan baru. Untuk menghapus aturan pemetaan, gunakan `DeleteServerMapping` perintah. Perintah ini menggunakan salah satu dari dua parameter berikut.

- `sourceTreePath`— jalur eksplisit ke objek database sumber Anda.
- `sourceNamePath`— jalur yang hanya mencakup nama-nama objek sumber Anda.

Untuk informasi selengkapnya tentang `DeleteServerMapping` perintah `AddServerMapping` dan, lihat [AWS Schema Conversion Tool CLI Referensi](#).

Langkah 3: Buat laporan penilaian

Sebelum Anda memulai migrasi, kami sarankan untuk membuat laporan penilaian. Laporan ini merangkum semua tugas migrasi dan merinci item tindakan yang akan muncul selama migrasi. Untuk memastikan migrasi Anda tidak gagal, lihat laporan ini dan alamat item tindakan sebelum migrasi. Untuk informasi selengkapnya, lihat [Laporan penilaian](#).

Pada langkah ini, Anda menggunakan `CreateMigrationReport` perintah. Perintah ini menggunakan dua parameter. `treePath` parameter-nya wajib, dan `forceMigrate` parameter-nya opsional.

- `treePath`— jalur eksplisit ke objek database sumber Anda yang Anda simpan salinan laporan penilaian.
- `forceMigrate`— ketika diatur ke `true`, AWS SCT melanjutkan migrasi bahkan jika proyek Anda menyertakan HDFS folder dan tabel Hive yang merujuk ke objek yang sama. Nilai default-nya adalah `false`.

Anda kemudian dapat menyimpan salinan laporan penilaian sebagai file value (CSV) yang dipisahkan koma PDF atau. Untuk melakukannya, gunakan `SaveReportCSV` perintah `SaveReportPDF` or.

`SaveReportPDF`Perintah menyimpan salinan laporan penilaian Anda sebagai PDF file. Perintah ini menggunakan empat parameter. `filePath` parameter-nya wajib, parameter lainnya bersifat opsional.

- `filePath`— jalur ke PDF file dan namanya.
- `filter`— nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.
- `treePath`— jalur eksplisit ke objek database sumber Anda yang Anda simpan salinan laporan penilaian.
- `namePath`— jalur yang hanya mencakup nama objek target Anda yang Anda simpan salinan laporan penilaian.

`SaveReportCSV`Perintah menyimpan laporan penilaian Anda dalam tiga CSV file. Perintah ini menggunakan empat parameter. `directory` parameter-nya wajib, parameter lainnya bersifat opsional.

- `directory`— jalur ke folder tempat AWS SCT menyimpan CSV file.

- `filter`— nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.
- `treePath`— jalur eksplisit ke objek database sumber Anda yang Anda simpan salinan laporan penilaian.
- `namePath`— jalur yang hanya mencakup nama objek target Anda yang Anda simpan salinan laporan penilaian.

Contoh kode berikut menyimpan salinan laporan penilaian dalam `c:\sct\ar.pdf` file.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

Contoh kode berikut menyimpan salinan laporan penilaian sebagai CSV file di `c:\sct` folder.

```
SaveReportCSV
-file:'c:\sct'
/
```

Untuk informasi selengkapnya tentang `SaveReportCSV` perintah `SaveReportPDF` dan, lihat [AWS Schema Conversion Tool CLI Referensi](#).

Langkah 4: Migrasikan cluster Apache Hadoop Anda ke Amazon dengan EMR AWS SCT

Setelah mengonfigurasi AWS SCT project, mulai migrasi klaster Apache Hadoop lokal Anda ke. AWS Cloud

Pada langkah ini, Anda menggunakan `Migrate`, `MigrationStatus`, dan `ResumeMigration` perintah.

`Migrate` Perintah memigrasikan objek sumber Anda ke cluster target. Perintah ini menggunakan empat parameter. Pastikan Anda menentukan `treePath` parameter `filter` atau. Parameter lainnya adalah opsional.

- `filter`— nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.

- `treePath`— jalur eksplisit ke objek database sumber Anda yang Anda simpan salinan laporan penilaian.
- `forceLoad`— ketika diatur ke `true`, AWS SCT secara otomatis memuat pohon metadata database selama migrasi. Nilai default-nya adalah `false`.
- `forceMigrate`— ketika diatur ke `true`, AWS SCT melanjutkan migrasi bahkan jika proyek Anda menyertakan HDFS folder dan tabel Hive yang merujuk ke objek yang sama. Nilai default-nya adalah `false`.

`MigrationStatusPerintah` mengembalikan informasi tentang kemajuan migrasi. Untuk menjalankan perintah ini, masukkan nama proyek migrasi Anda untuk `name` parameter. Anda menentukan nama ini dalam `CreateProject` perintah.

`ResumeMigrationPerintah` melanjutkan migrasi terputus yang Anda luncurkan menggunakan perintah. `Migrate ResumeMigrationPerintah` tidak menggunakan parameter. Untuk melanjutkan migrasi, Anda harus terhubung ke sumber dan kluster target Anda. Untuk informasi selengkapnya, lihat [Mengelola proyek migrasi](#).

Contoh kode berikut memigrasikan data dari HDFS layanan sumber Anda ke AmazonEMR.

```
Migrate
-treePath: 'Clusters.HADOOP_SOURCE.HDFS_SOURCE'
-forceMigrate: 'true'
/
```

Menjalankan CLI skrip Anda

Setelah Anda selesai mengedit AWS SCT CLI skrip Anda, simpan sebagai file dengan `.scts` ekstensi. Sekarang, Anda dapat menjalankan skrip Anda dari `app` folder jalur AWS SCT instalasi Anda. Untuk melakukannya, gunakan perintah berikut.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\hadoop.scts"
```

Pada contoh sebelumnya, ganti `script_path` dengan jalur ke file Anda dengan CLI skrip. Untuk informasi selengkapnya tentang menjalankan CLI skrip AWS SCT, lihat [Mode skrip](#).

Mengelola proyek migrasi data besar Anda

Setelah menyelesaikan migrasi, Anda dapat menyimpan dan mengedit AWS SCT proyek Anda untuk digunakan di masa mendatang.

Untuk menyimpan AWS SCT proyek Anda, gunakan `SaveProject` perintah. Perintah ini tidak menggunakan parameter.

Contoh kode berikut menyimpan AWS SCT proyek Anda.

```
SaveProject  
/
```

Untuk membuka AWS SCT proyek Anda, gunakan `OpenProject` perintah. Perintah ini menggunakan satu parameter wajib. Untuk `file` parameter, masukkan path ke file AWS SCT proyek Anda dan namanya. Anda menentukan nama proyek dalam `CreateProject` perintah. Pastikan Anda menambahkan `.scts` ekstensi ke nama file proyek Anda untuk menjalankan `OpenProject` perintah.

Contoh kode berikut membuka `hadoop_emr` proyek dari `c:\sct` folder.

```
OpenProject  
-file: 'c:\sct\hadoop_emr.scts'  
/
```

Setelah Anda membuka AWS SCT proyek Anda, Anda tidak perlu menambahkan sumber dan target cluster karena Anda telah menembarkannya ke proyek Anda. Untuk mulai bekerja dengan sumber dan kluster target Anda, Anda harus terhubung dengan mereka. Untuk melakukannya, Anda menggunakan `ConnectTargetCluster` perintah `ConnectSourceCluster` dan. Perintah ini menggunakan parameter yang sama dengan `AddTargetCluster` perintah `AddSourceCluster` dan. Anda dapat mengedit CLI skrip Anda dan mengganti nama perintah ini meninggalkan daftar parameter tanpa perubahan.

Contoh kode berikut menghubungkan ke cluster Hadoop sumber.

```
ConnectSourceCluster  
-name: 'HADOOP_SOURCE'  
-vendor: 'HADOOP'  
-host: 'hadoop_address'
```

```
-port: '22'  
-user: 'hadoop_user'  
-password: 'hadoop_password'  
-useSSL: 'true'  
-privateKeyPath: 'c:\path\name.pem'  
-passPhrase: 'hadoop_passphrase'
```

/

Contoh kode berikut terhubung ke EMR cluster Amazon target.

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

/

Pada contoh sebelumnya, ganti *hadoop_address* dengan alamat IP cluster Hadoop Anda. Jika diperlukan, konfigurasi nilai variabel port. Selanjutnya, ganti *hadoop_user* and *hadoop_password* dengan nama pengguna Hadoop Anda dan kata sandi untuk pengguna ini. Untuk *path\name*, masukkan nama dan path ke PEM file untuk cluster Hadoop sumber Anda. Untuk informasi selengkapnya tentang menambahkan sumber dan kluster target, lihat [Menghubungkan ke database Apache Hadoop dengan AWS Schema Conversion Tool](#).

Setelah Anda terhubung ke sumber dan target cluster Hadoop, Anda harus terhubung ke Hive dan HDFS layanan Anda, serta ke bucket Amazon S3 Anda. Untuk melakukannya, Anda menggunakan `ConnectSourceClusterHive`, `ConnectSourceClusterHdfs`, `ConnectTargetClusterHive`, `ConnectTargetClusterHdfs`, dan `ConnectTargetClusterS3` perintah. Perintah ini menggunakan parameter yang sama dengan perintah yang Anda gunakan untuk menambahkan Hive dan HDFS layanan, serta bucket Amazon S3 ke project Anda. Edit CLI skrip untuk mengganti Add awalan Connect dengan nama perintah.

Mengonversi workflowas Oozie menjadi dengan AWS Step FunctionsAWS Schema Conversion Tool

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

Topik

- [Ikhtisar konversi](#)
- [Langkah 1: Connect ke layanan sumber dan target Anda](#)
- [Langkah 2: Siapkan aturan pemetaan](#)
- [Langkah 3: Konfigurasi parameter](#)
- [Langkah 4: Buat laporan penilaian](#)
- [Langkah 5: Konversikan alur kerja Apache Oozie Anda ke dengan AWS Step FunctionsAWS SCT](#)
- [Menjalankan CLI skrip Anda](#)
- [Apache Oozie node yang AWS SCT dapat dikonversi ke AWS Step Functions](#)

Ikhtisar konversi

Kode sumber Apache Oozie Anda mencakup node tindakan, node aliran kontrol, dan properti pekerjaan. Action node menentukan pekerjaan, yang Anda jalankan dalam alur kerja Apache Oozie Anda. Ketika Anda menggunakan Apache Oozie untuk mengatur cluster Apache Hadoop Anda, maka node tindakan menyertakan pekerjaan Hadoop. Node aliran kontrol menyediakan mekanisme untuk mengontrol jalur alur kerja. Node aliran kontrol termasuk node seperti `start`, `end`, `decision`, `fork`, dan `join`.

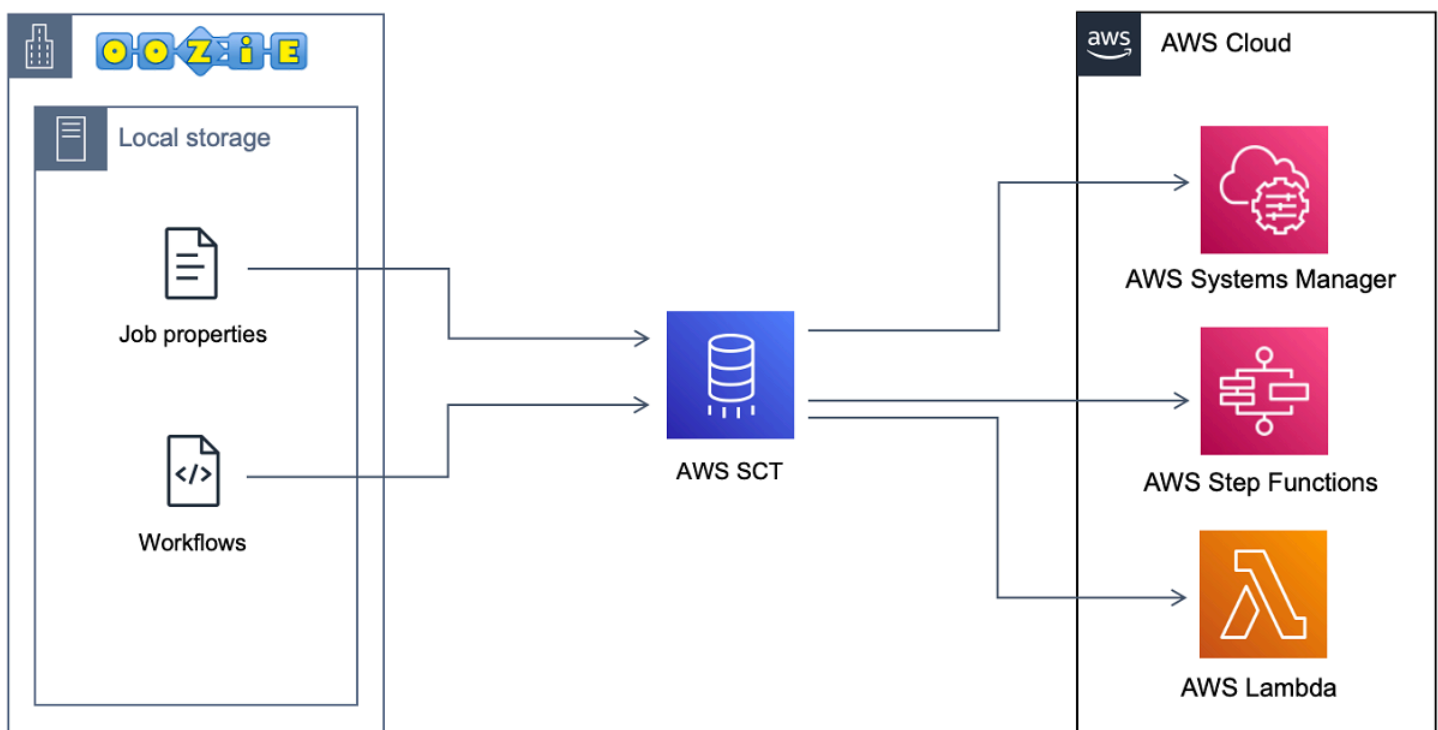
AWS SCT mengonversi node aksi sumber Anda dan mengontrol node aliran menjadi AWS Step Functions. Di AWS Step Functions, Anda menentukan alur kerja Anda di Amazon States Language (ASL). AWS SCT digunakan ASL untuk mendefinisikan mesin status Anda, yang merupakan kumpulan status, yang dapat melakukan pekerjaan, menentukan status mana yang akan dialihkan ke berikutnya, berhenti dengan kesalahan, dan sebagainya. Selanjutnya, AWS SCT unggah JSON file dengan definisi mesin negara. Kemudian, AWS SCT dapat menggunakan peran AWS Identity and Access Management (IAM) Anda untuk mengonfigurasi mesin status Anda AWS Step Functions. Untuk informasi lebih lanjut, lihat [Apa itu AWS Step Functions?](#) di Panduan AWS Step Functions Pengembang.

Juga, AWS SCT buat paket ekstensi dengan AWS Lambda fungsi yang meniru fungsi sumber yang AWS Step Functions tidak mendukung. Untuk informasi selengkapnya, lihat [Menggunakan paket ekstensi dengan AWS Schema Conversion Tool](#).

AWS SCT memigrasikan properti pekerjaan sumber Anda ke AWS Systems Manager. Untuk menyimpan nama dan nilai parameter, AWS SCT gunakan Parameter Store, kemampuan AWS Systems Manager. Untuk informasi lebih lanjut, lihat [Apa itu AWS Systems Manager?](#) dalam AWS Systems Manager User Guide.

Anda dapat menggunakan AWS SCT untuk memperbarui nilai dan nama parameter Anda secara otomatis. Karena perbedaan arsitektur antara Apache Oozie dan AWS Step Functions, Anda mungkin perlu mengkonfigurasi parameter Anda. AWS SCT dapat menemukan nama parameter tertentu atau nilai dalam file sumber Anda dan menggantinya dengan nilai-nilai baru. Untuk informasi selengkapnya, lihat [Langkah 3: Konfigurasi parameter](#).

Gambar berikut menunjukkan diagram arsitektur konversi Apache Oozie ke AWS Step Functions



Untuk memulai konversi, buat dan jalankan AWS SCT CLI skrip Anda. Skrip ini mencakup set lengkap perintah untuk menjalankan konversi. Anda dapat mengunduh dan mengedit templat skrip konversi Apache Oozie. Untuk informasi selengkapnya, lihat [Mendapatkan CLI skenario](#).

Pastikan skrip Anda menyertakan langkah-langkah berikut.

Langkah 1: Connect ke layanan sumber dan target Anda

Untuk memulai konversi cluster Apache Oozie Anda, buat proyek baru. AWS SCT Selanjutnya, sambungkan ke layanan sumber dan target Anda. Pastikan Anda membuat dan menyediakan AWS sumber daya target sebelum memulai migrasi. Untuk informasi selengkapnya, lihat [Prasyarat untuk menggunakan Apache Oozie sebagai sumber](#).

Pada langkah ini, Anda menggunakan AWS SCT CLI perintah berikut.

- `CreateProject`— untuk membuat AWS SCT proyek baru.
- `AddSource`— untuk menambahkan file Apache Oozie sumber Anda dalam proyek Anda. AWS SCT
- `ConnectSource`— untuk terhubung ke Apache Oozie sebagai sumber.
- `AddTarget`— untuk menambahkan AWS Step Functions sebagai target migrasi dalam proyek Anda.
- `ConnectTarget`— untuk terhubung ke AWS Step Functions.

Untuk contoh menggunakan AWS SCT CLI perintah ini, lihat [Menghubungkan ke Apache Oozie](#).

Ketika Anda menjalankan `ConnectTarget` perintah `ConnectSource` atau, AWS SCT mencoba untuk membuat koneksi ke layanan Anda. Jika upaya koneksi gagal, maka AWS SCT berhenti menjalankan perintah dari CLI skrip Anda dan menampilkan pesan kesalahan.

Langkah 2: Siapkan aturan pemetaan

Setelah Anda terhubung ke layanan sumber dan target, atur aturan pemetaan. Aturan pemetaan menentukan target migrasi untuk alur kerja dan parameter Apache Oozie sumber Anda. Untuk informasi selengkapnya tentang aturan pemetaan, lihat [Memetakan tipe data di AWS Schema Conversion Tool](#).

Untuk menentukan sumber dan objek target untuk konversi, gunakan `AddServerMapping` perintah. Perintah ini menggunakan dua parameter: `sourceTreePath` dan `targetTreePath`. Nilai parameter ini mencakup jalur eksplisit ke objek sumber dan target Anda. Untuk Apache Oozie ke AWS Step Functions konversi, parameter ini harus dimulai dengan. ETL

Contoh kode berikut membuat aturan pemetaan untuk OOOZIE dan AWS_STEP_FUNCTIONS objek. Anda menambahkan objek ini ke AWS SCT proyek Anda menggunakan `AddSource` dan `AddTarget` perintah di langkah sebelumnya.

```
AddServerMapping
  -sourceTreePath: 'ETL.APACHE_OOZIE'
  -targetTreePath: 'ETL.AWS_STEP_FUNCTIONS'
/
```

Untuk informasi selengkapnya tentang `AddServerMapping` perintah, lihat [AWS Schema Conversion Tool CLI Referensi](#).

Langkah 3: Konfigurasi parameter

Jika alur kerja Apache Oozie sumber Anda menggunakan parameter, Anda mungkin perlu mengubah nilainya setelah konversi ke AWS Step Functions. Juga, Anda mungkin perlu menambahkan parameter baru untuk digunakan dengan AWS Step Functions.

Pada langkah ini, Anda menggunakan `AddTargetParameter` perintah `AddParameterMapping` dan.

Untuk mengganti nilai parameter dalam file sumber Anda, gunakan `AddParameterMapping` perintah. AWS SCT memindai file sumber Anda, menemukan parameter berdasarkan nama atau nilai, dan mengubah nilainya. Anda dapat menjalankan satu perintah untuk memindai semua file sumber Anda. Anda menentukan ruang lingkup file yang akan dipindai menggunakan salah satu dari tiga parameter pertama dari daftar berikut. Perintah ini menggunakan hingga enam parameter.

- `filterName`— nama filter untuk objek sumber Anda. Anda dapat membuat filter menggunakan `CreateFilter` perintah.
- `treePath`— jalur eksplisit ke objek sumber Anda.
- `namePath`— jalur eksplisit ke objek sumber tertentu.
- `sourceParameterName`— nama parameter sumber Anda.
- `sourceValue`— nilai parameter sumber Anda.
- `targetValue`— nilai parameter target Anda.

Contoh kode berikut menggantikan semua parameter di mana nilai sama `c:\oozie\hive.py` dengan `s3://bucket-oozie/hive.py` nilai.

```
AddParameterMapping
  -treePath: 'ETL.OOZIE.Applications'
  -sourceValue: 'c:\oozie\hive.py'
```

```
-targetValue: 's3://bucket-oozie/hive.py'  
/
```

Contoh kode berikut menggantikan semua parameter di mana nama sama nameNode dengan `hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020` nilai.

```
AddParameterMapping  
-treePath: 'ETL.OOZIE_SOURCE.Applications'  
-sourceParameter: 'nameNode'  
-targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'  
/
```

Contoh kode berikut menggantikan semua parameter di mana nama sama dengan nameNode dan nilainya sama `hdfs://ip-55.eu-west-1.compute.internal:8020` dengan nilai dari `targetValue` parameter.

```
AddParameterMapping  
-treePath: 'ETL.OOZIE_SOURCE.Applications'  
-sourceParameter: 'nameNode'  
-sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'  
-targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'  
/
```

Untuk menambahkan parameter baru dalam file target Anda selain parameter yang ada dari file sumber Anda, gunakan `AddTargetParameter` perintah. Perintah ini menggunakan set parameter yang sama dengan `AddParameterMapping` perintah.

Contoh kode berikut menambahkan parameter `clusterId` target bukan nameNode parameter.

```
AddTargetParameter  
-treePath: 'ETL.OOZIE_SOURCE.Applications'  
-sourceParameter: 'nameNode'  
-sourceValue: 'hdfs://ip-55-66-77-88.eu-west-1.compute.internal:8020'  
-targetParameter: 'clusterId'  
-targetValue: '1234567890abcdef0'  
/
```

Untuk informasi selengkapnya tentang `CreateFilter` perintah

`AddServerMapping`, `AddParameterMapping`, `AddTargetParameter`, dan, lihat [AWS Schema Conversion Tool CLI Referensi](#).

Langkah 4: Buat laporan penilaian

Sebelum Anda memulai konversi, kami sarankan untuk membuat laporan penilaian. Laporan ini merangkum semua tugas migrasi dan merinci item tindakan yang akan muncul selama migrasi. Untuk memastikan migrasi Anda tidak gagal, lihat laporan ini dan alamat item tindakan sebelum migrasi. Untuk informasi selengkapnya, lihat [Laporan penilaian](#).

Pada langkah ini, Anda menggunakan `CreateReport` perintah. Perintah ini menggunakan dua parameter. Parameter pertama menjelaskan objek sumber yang AWS SCT membuat laporan penilaian. Untuk melakukannya, gunakan salah satu parameter berikut: `filterName`, `treePath`, atau `namePath`. Parameter ini wajib. Juga, Anda dapat menambahkan parameter `forceLoad` Boolean opsional. Jika Anda mengatur parameter `init` `true`, maka AWS SCT secara otomatis memuat semua objek anak untuk objek sumber yang Anda tentukan dalam `CreateReport` perintah.

Contoh kode berikut membuat laporan penilaian untuk `Applications` node file Oozie sumber Anda.

```
CreateReport
  -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

Anda kemudian dapat menyimpan salinan laporan penilaian sebagai file value (CSV) yang dipisahkan koma PDF atau. Untuk melakukannya, gunakan `SaveReportCSV` perintah `SaveReportPDF` or.

`SaveReportPDF`Perintah menyimpan salinan laporan penilaian Anda sebagai PDF file. Perintah ini menggunakan empat parameter. `filePath`nya wajib, parameter lainnya bersifat opsional.

- `filePath`— jalur ke PDF file dan namanya.
- `filter`— nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.
- `treePath`— jalur eksplisit ke objek database sumber Anda yang Anda simpan salinan laporan penilaian.
- `namePath`— jalur yang hanya mencakup nama objek target Anda yang Anda simpan salinan laporan penilaian.

`SaveReportCSV`Perintah menyimpan laporan penilaian Anda dalam CSV file. Perintah ini menggunakan empat parameter. `directoryPath`nya wajib, parameter lainnya bersifat opsional.

- `directory`— jalur ke folder tempat AWS SCT menyimpan CSV file.
- `filter`— nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.
- `treePath`— jalur eksplisit ke objek database sumber Anda yang Anda simpan salinan laporan penilaian.
- `namePath`— jalur yang hanya mencakup nama objek target Anda yang Anda simpan salinan laporan penilaian.

Contoh kode berikut menyimpan salinan laporan penilaian dalam `c:\sct\ar.pdf` file.

```
SaveReportPDF
-file:'c:\sct\ar.pdf'
/
```

Contoh kode berikut menyimpan salinan laporan penilaian sebagai CSV file di `c:\sct` folder.

```
SaveReportCSV
-file:'c:\sct'
/
```

Untuk informasi selengkapnya tentang `CreateReport`, `SaveReportPDF` dan `SaveReportCSV` perintah, lihat [AWS Schema Conversion Tool CLI Referensi](#).

Langkah 5: Konversikan alur kerja Apache Oozie Anda ke dengan AWS Step Functions AWS SCT

Setelah Anda mengonfigurasi AWS SCT proyek Anda, konversikan kode sumber Anda dan terapkan ke file AWS Cloud.

Pada langkah ini, Anda menggunakan `Convert`, `SaveOnS3ConfigureStateMachine`, dan `ApplyToTarget` perintah.

`MigratePerintah` memigrasikan objek sumber Anda ke cluster target. Perintah ini menggunakan empat parameter. Pastikan Anda menentukan `treePath` parameter `filter` atau. Parameter lainnya adalah opsional.

- `filter`— nama filter yang Anda buat sebelumnya untuk menentukan ruang lingkup objek sumber Anda untuk bermigrasi.

- `namePath`— jalur eksplisit ke objek sumber tertentu.
- `treePath`— jalur eksplisit ke objek database sumber Anda yang Anda simpan salinan laporan penilaian.
- `forceLoad`— ketika diatur ke `true`, AWS SCT secara otomatis memuat pohon metadata database selama migrasi. Nilai default-nya adalah `false`.

Contoh kode berikut mengonversi file dari `Applications` folder di file Oozie sumber Anda.

```
Convert
  -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

`SaveOnS3` Mengunggah definisi mesin status ke bucket Amazon S3 Anda. Perintah ini menggunakan `treePath` parameter. Untuk menjalankan perintah ini, gunakan folder target dengan definisi mesin status sebagai nilai parameter ini.

Berikut ini mengunggah State machine definitions folder objek `AWS_STEP_FUNCTIONS` target Anda ke bucket Amazon S3. AWS SCT menggunakan bucket Amazon S3 yang Anda simpan di profil AWS layanan di langkah tersebut. [Prasyarat](#)

```
SaveOnS3
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
/
```

`ConfigureStateMachine` Perintah mengkonfigurasi mesin negara. Perintah ini menggunakan hingga enam parameter. Pastikan Anda menentukan cakupan target menggunakan salah satu dari tiga parameter pertama dari daftar berikut.

- `filterName`— nama filter untuk objek target Anda. Anda dapat membuat filter menggunakan `CreateFilter` perintah.
- `treePath`— jalur eksplisit ke objek target Anda.
- `namePath`— jalur eksplisit ke objek target tertentu.
- `iamRole`— Nama Sumber Daya Amazon (ARN) dari IAM peran yang menyediakan akses ke mesin langkah Anda. Parameter ini diperlukan.

Contoh kode berikut mengkonfigurasi mesin negara yang didefinisikan dalam `AWS_STEP_FUNCTIONS` menggunakan `role_name` IAM peran.

```
ConfigureStateMachine
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
  -role: 'arn:aws:iam::555555555555:role/role_name'
/
```

ApplyToTargetPerintah tersebut menerapkan kode konversi Anda ke server target. Untuk menjalankan perintah ini, gunakan salah satu parameter berikut: `filterName`, `treePath`, atau `namePath` untuk menentukan objek target untuk diterapkan.

Contoh kode berikut menerapkan mesin `app_wp` status ke AWS Step Functions.

```
ApplyToTarget
  -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machines.app_wp'
/
```

Untuk memastikan bahwa kode yang dikonversi menghasilkan hasil yang sama dengan kode sumber Anda, Anda dapat menggunakan paket AWS SCT ekstensi. Ini adalah seperangkat AWS Lambda fungsi yang meniru fungsi Apache Oozie Anda yang AWS Step Functions tidak mendukung. Untuk menginstal paket ekstensi ini, Anda dapat menggunakan `CreateLambdaExtPack` perintah.

Perintah ini menggunakan hingga lima parameter. Pastikan Anda menggunakannya **Oozie2SF** untuk `extPackId`. Dalam hal ini, AWS SCT membuat paket ekstensi untuk sumber fungsi Apache Oozie.

- `extPackId`— pengenal unik untuk satu set fungsi Lambda. Parameter ini diperlukan.
- `tempDirectory`— jalur di mana AWS SCT dapat menyimpan file sementara. Parameter ini diperlukan.
- `awsProfile`— nama AWS profil Anda.
- `lambdaExecRoles`— daftar Nama Sumber Daya Amazon (ARNs) dari peran eksekusi yang akan digunakan untuk fungsi Lambda.
- `createInvokeRoleFlag`— bendera Boolean yang menunjukkan apakah akan membuat peran eksekusi untuk AWS Step Functions.

Untuk menginstal dan menggunakan paket ekstensi, pastikan Anda memberikan izin yang diperlukan. Untuk informasi selengkapnya, lihat [Izin untuk menggunakan AWS Lambda fungsi dalam paket ekstensi](#).

Untuk informasi selengkapnya tentang CreateLambdaExtPack perintah ConvertSaveOnS3,ConfigureStateMachine,ApplyToTarget,, dan, lihat [AWS Schema Conversion Tool CLIReferensi](#).

Menjalankan CLI skrip Anda

Setelah Anda selesai mengedit AWS SCT CLI skrip Anda, simpan sebagai file dengan .scts ekstensi. Sekarang, Anda dapat menjalankan skrip Anda dari app folder jalur AWS SCT instalasi Anda. Untuk melakukannya, gunakan perintah berikut.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\oozie.scts"
```

Pada contoh sebelumnya, ganti *script_path* dengan jalur ke file Anda dengan CLI skrip. Untuk informasi selengkapnya tentang menjalankan CLI skrip AWS SCT, lihat [Mode skrip](#).

Apache Oozie node yang AWS SCT dapat dikonversi ke AWS Step Functions

Anda dapat menggunakan AWS SCT untuk mengonversi node aksi Apache Oozie dan mengontrol node aliran ke. AWS Step Functions

Node tindakan yang didukung meliputi:

- Tindakan sarang
- Aksi Hive2
- Aksi percikan
- MapReduce Aksi streaming
- Tindakan Java
- DistCp aksi
- Aksi babi
- Tindakan Sqoop
- Tindakan FS
- Tindakan Shell

Node aliran kontrol yang didukung meliputi:

- Mulai tindakan
- Akhiri tindakan
- Bunuh aksi
- Tindakan keputusan
- Aksi garpu
- Bergabunglah dengan aksi

Integrasi dengan AWS Database Migration Service

AWS Schema Conversion Tool

Menggunakan agen AWS SCT replikasi dengan AWS DMS

Untuk migrasi database yang sangat besar, Anda dapat menggunakan agen AWS SCT replikasi (`aws-schema-conversion-tool-dms-agent`) untuk menyalin data dari database lokal ke Amazon S3 atau perangkat Edge. AWS Snowball Agen replikasi bekerja bersama dengan AWS DMS dan dapat bekerja di latar belakang saat AWS SCT ditutup.

Saat bekerja dengan AWS Snowball Edge, AWS SCT agen mereplikasi data ke AWS Snowball perangkat. Perangkat kemudian dikirim ke AWS dan data dimuat ke ember Amazon S3. Selama waktu ini, AWS SCT agen terus berjalan. Agen kemudian mengambil data di Amazon S3 dan menyalin data ke titik akhir target.

Untuk informasi selengkapnya, lihat [Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool](#).

Menggunakan agen ekstraksi AWS SCT data dengan AWS DMS

Di AWS SCT, Anda dapat menemukan agen ekstraksi data (`aws-schema-conversion-tool-extractor`) yang membantu memudahkan migrasi dari Apache Cassandra ke Amazon DynamoDB. Cassandra dan DynamoDB adalah Tidak ada SQL database, tetapi mereka berbeda dalam arsitektur sistem dan representasi data. Anda dapat menggunakan alur kerja berbasis wizard untuk mengotomatiskan proses migrasi AWS SCT Cassandra-ke-DynamoDB. AWS SCT terintegrasi dengan AWS Database Migration Service (AWS DMS) untuk melakukan migrasi yang sebenarnya.

Untuk informasi selengkapnya, lihat [Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool](#).

Meningkatkan level logging saat menggunakan AWS SCT dengan AWS DMS

Anda dapat meningkatkan level logging saat menggunakan AWS SCT dengan AWS DMS, misalnya jika Anda perlu bekerja dengan AWS Support.

Setelah menginstal AWS SCT dan driver yang diperlukan, buka aplikasi dengan memilih AWS SCT ikon. Jika Anda melihat pemberitahuan pembaruan, Anda dapat memilih untuk memperbarui sebelum atau setelah proyek Anda selesai. Jika jendela proyek otomatis terbuka, tutup jendela dan buat proyek secara manual.

Untuk meningkatkan level logging saat menggunakan AWS SCT dengan AWS DMS

1. Pada menu Pengaturan, pilih Pengaturan global.
2. Di jendela Pengaturan global, pilih Logging.
3. Untuk mode Debug, pilih True.
4. Dari bagian Tingkat pesan, Anda dapat memodifikasi jenis log berikut:
 - Umum
 - Loader
 - Parser
 - Printer
 - Penyelesai
 - Telemetri
 - Konverter

Secara default, semua tingkat pesan diatur ke Info.

5. Pilih tingkat pencatatan untuk jenis tingkat pesan apa pun yang ingin Anda ubah:
 - Jejak (pencatatan paling detail)
 - Debug
 - Info
 - Peringatan
 - Kesalahan (logging paling tidak rinci)
 - Kritis
 - Wajib
6. Pilih Terapkan untuk mengubah pengaturan untuk proyek Anda.
7. Pilih OK untuk menutup jendela pengaturan Global.

Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool

Anda dapat menggunakan AWS SCT agen untuk mengekstrak data dari gudang data lokal dan memigrasikannya ke Amazon Redshift. Agen mengekstrak data Anda dan mengunggah data ke Amazon S3 atau, untuk migrasi skala besar, perangkat Edge. AWS Snowball Anda kemudian dapat menggunakan AWS SCT agen untuk menyalin data ke Amazon Redshift.

Atau, Anda dapat menggunakan AWS Database Migration Service (AWS DMS) untuk memigrasikan data ke Amazon Redshift. AWS DMS Keuntungannya adalah dukungan replikasi yang sedang berlangsung (perubahan pengambilan data). Namun, untuk meningkatkan kecepatan migrasi data, gunakan beberapa AWS SCT agen secara paralel. Menurut pengujian kami, AWS SCT agen memigrasikan data lebih cepat daripada AWS DMS 15-35 persen. Perbedaan kecepatan adalah karena kompresi data, dukungan migrasi partisi tabel secara paralel, dan pengaturan konfigurasi yang berbeda. Untuk informasi selengkapnya, lihat [Menggunakan database Amazon Redshift sebagai target](#). AWS Database Migration Service

Amazon S3 adalah layanan penyimpanan dan pengambilan. Untuk menyimpan objek di Amazon S3, Anda mengunggah file yang ingin Anda simpan ke bucket Amazon S3. Saat Anda mengunggah file, Anda dapat mengatur izin pada objek dan juga pada metadata apa pun.

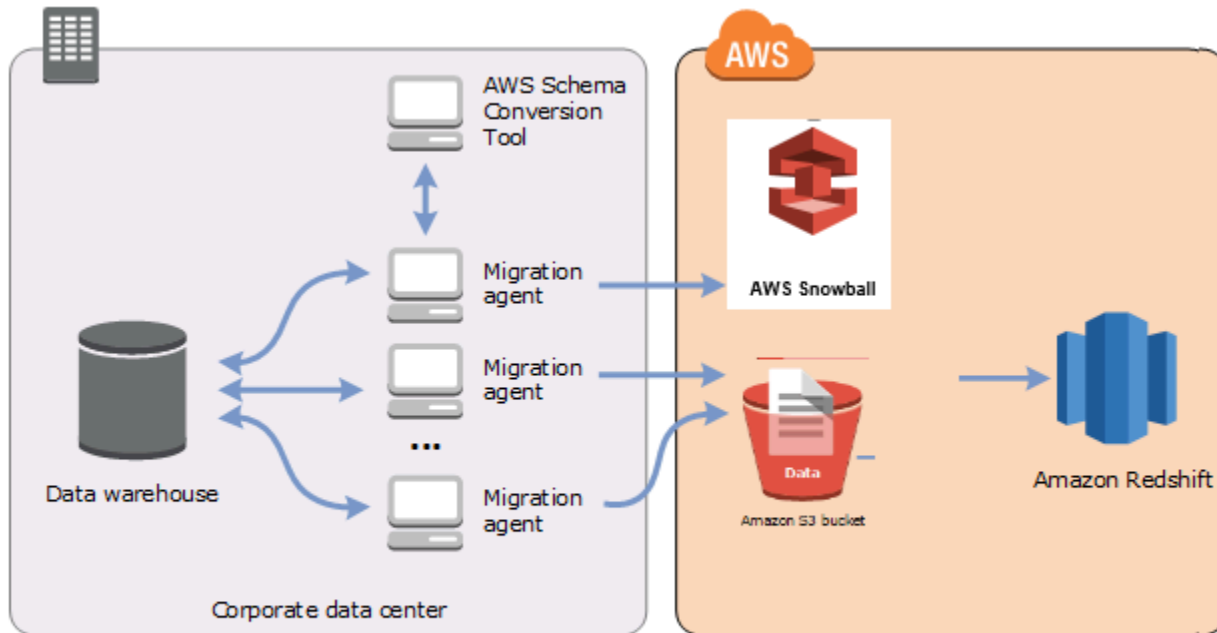
Migrasi skala besar

Migrasi data skala besar dapat mencakup banyak terabyte informasi, dan dapat diperlambat oleh kinerja jaringan dan oleh banyaknya data yang harus dipindahkan. AWS Snowball Edge adalah AWS layanan yang dapat Anda gunakan untuk mentransfer data ke cloud dengan faster-than-network kecepatan menggunakan alat yang AWS dimiliki. Perangkat AWS Snowball Edge dapat menampung hingga 100 TB data. Menggunakan enkripsi 256-bit dan Trusted Platform Module (TPM) standar industri untuk memastikan keamanan dan penuh chain-of-custody untuk data Anda. AWS SCT bekerja dengan perangkat AWS Snowball Edge.

Saat Anda menggunakan AWS SCT dan perangkat AWS Snowball Edge, Anda memigrasikan data dalam dua tahap. Pertama, Anda gunakan AWS SCT untuk memproses data secara lokal dan kemudian memindahkan data itu ke perangkat AWS Snowball Edge. Anda kemudian mengirim perangkat untuk AWS menggunakan proses AWS Snowball Edge, dan kemudian AWS secara otomatis memuat data ke dalam bucket Amazon S3. Selanjutnya, ketika data tersedia di Amazon S3,

Anda gunakan AWS SCT untuk memigrasikan data ke Amazon Redshift. Agen ekstraksi data dapat bekerja di latar belakang saat AWS SCT ditutup.

Diagram berikut menunjukkan skenario yang didukung.



Agen ekstraksi data saat ini didukung untuk gudang data sumber berikut:

- Analisis Sinaps Azure
- BigQuery
- Database Greenplum (versi 4.3)
- Microsoft SQL Server (versi 2008 dan lebih tinggi)
- Netezza (versi 7.0.3 dan lebih tinggi)
- Oracle (versi 10 dan lebih tinggi)
- Kepingan salju (versi 3)
- Teradata (versi 13 dan lebih tinggi)
- Vertica (versi 7.2.2 dan lebih tinggi)

Anda dapat terhubung ke FIPS titik akhir untuk Amazon Redshift jika Anda perlu mematuhi persyaratan keamanan Federal Information Processing Standard FIPS (). FIPSendpoint tersedia di AWS Wilayah berikut:

- Wilayah AS Timur (Virginia N.) (redshift-fips.us-east-1.amazonaws.com)
- Wilayah Timur AS (Ohio) (redshift-fips.us-east-2.amazonaws.com)
- Wilayah AS Barat (California N.) (redshift-fips.us-west-1.amazonaws.com)
- Wilayah Barat AS (Oregon) (redshift-fips.us-west-2.amazonaws.com)

Gunakan informasi dalam topik berikut untuk mempelajari cara bekerja dengan agen ekstraksi data.

Topik

- [Prasyarat untuk menggunakan agen ekstraksi data](#)
- [Instalasi agen ekstraksi](#)
- [Mengkonfigurasi agen ekstraksi](#)
- [Mendaftarkan agen ekstraksi dengan AWS Schema Conversion Tool](#)
- [Menyembunyikan dan memulihkan informasi untuk agen AWS SCT](#)
- [Membuat aturan migrasi data di AWS SCT](#)
- [Mengubah pengaturan ekstraktor dan salin dari pengaturan proyek](#)
- [Menyortir data sebelum bermigrasi menggunakan AWS SCT](#)
- [Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data](#)
- [Mengekspor dan mengimpor tugas ekstraksi AWS SCT data](#)
- [Ekstraksi data menggunakan perangkat AWS Snowball Edge](#)
- [Output tugas ekstraksi data](#)
- [Menggunakan partisi virtual dengan AWS Schema Conversion Tool](#)
- [Menggunakan partisi asli](#)
- [Bermigrasi LOBs ke Amazon Redshift](#)
- [Praktik terbaik dan pemecahan masalah untuk agen ekstraksi data](#)

Prasyarat untuk menggunakan agen ekstraksi data

Sebelum Anda bekerja dengan agen ekstraksi data, tambahkan izin yang diperlukan untuk Amazon Redshift sebagai target ke pengguna Amazon Redshift Anda. Untuk informasi selengkapnya, lihat [Izin untuk Amazon Redshift sebagai target](#).

Kemudian, simpan informasi bucket Amazon S3 Anda dan siapkan kepercayaan Secure Sockets Layer (SSL) dan penyimpanan kunci Anda.

Pengaturan Amazon S3

Setelah agen Anda mengekstrak data Anda, mereka mengunggahnya ke bucket Amazon S3 Anda. Sebelum melanjutkan, Anda harus memberikan kredensial untuk terhubung ke AWS akun Anda dan bucket Amazon S3 Anda. Anda menyimpan kredensial dan informasi bucket Anda di profil di pengaturan aplikasi global, lalu mengaitkan profil dengan proyek Anda AWS SCT . Jika perlu, pilih Pengaturan global untuk membuat profil baru. Untuk informasi selengkapnya, lihat [Mengelola Profil di AWS Schema Conversion Tool](#).

Untuk memigrasikan data ke database Amazon Redshift target, AWS SCT agen ekstraksi data memerlukan izin untuk mengakses bucket Amazon S3 atas nama Anda. Untuk memberikan izin ini, buat pengguna AWS Identity and Access Management (IAM) dengan kebijakan berikut.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:GetObject",
        "s3:GetObjectTagging",
        "s3:PutObjectTagging"
      ],
      "Resource": [
        "arn:aws:s3:::bucket_name/*",
        "arn:aws:s3:::bucket_name"
      ],
      "Effect": "Allow"
    },
    {
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::bucket_name"
      ],
      "Effect": "Allow"
    },
    {
      "Effect": "Allow",
```

```
        "Action": "s3:ListAllMyBuckets",
        "Resource": "*"
    },
    {
        "Action": [
            "iam:GetUser"
        ],
        "Resource": [
            "arn:aws:iam::111122223333:user/DataExtractionAgentName"
        ],
        "Effect": "Allow"
    }
]
```

Pada contoh sebelumnya, ganti *bucket_name* dengan nama bucket Amazon S3 Anda. Kemudian, ganti *111122223333:user/DataExtractionAgentName* dengan nama IAM pengguna Anda.

Dengan asumsi peran IAM

Untuk keamanan tambahan, Anda dapat menggunakan AWS Identity and Access Management (IAM) peran untuk mengakses bucket Amazon S3. Untuk melakukannya, buat IAM pengguna untuk agen ekstraksi data Anda tanpa izin apa pun. Kemudian, buat IAM peran yang memungkinkan akses Amazon S3, dan tentukan daftar layanan dan pengguna yang dapat mengambil peran ini. Untuk informasi selengkapnya, lihat [IAMperan](#) di Panduan IAM Pengguna.

Untuk mengonfigurasi IAM peran untuk mengakses bucket Amazon S3 Anda

1. Buat IAM pengguna baru. Untuk kredensi pengguna, pilih Jenis akses terprogram.
2. Konfigurasi lingkungan host sehingga agen ekstraksi data Anda dapat mengambil peran yang AWS SCT menyediakan. Pastikan bahwa pengguna yang Anda konfigurasi pada langkah sebelumnya memungkinkan agen ekstraksi data untuk menggunakan rantai penyedia kredensi. Untuk informasi selengkapnya, lihat [Menggunakan kredensial](#) di Panduan AWS SDK for Java Pengembang.
3. Buat IAM peran baru yang memiliki akses ke bucket Amazon S3 Anda.
4. Ubah bagian kepercayaan dari peran ini untuk mempercayai pengguna yang Anda buat sebelumnya untuk mengambil peran tersebut. Dalam contoh berikut, ganti *111122223333:user/DataExtractionAgentName* dengan nama pengguna Anda.

```
{
```



```
"Effect": "Allow",
"Principal": {
  "AWS": "arn:aws:iam::111122223333:user/DataExtractionAgentName"
},
"Action": "sts:AssumeRole"
}
```

- Ubah bagian kepercayaan dari peran ini menjadi kepercayaan `redshift.amazonaws.com` untuk mengambil peran.

```
{
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "redshift.amazonaws.com"
    ]
  },
  "Action": "sts:AssumeRole"
}
```

- Lampirkan peran ini ke cluster Amazon Redshift Anda.

Sekarang, Anda dapat menjalankan agen ekstraksi data Anda di AWS SCT.

Bila Anda menggunakan asumsi IAM peran, migrasi data berfungsi dengan cara berikut. Agen ekstraksi data memulai dan mendapatkan kredensial pengguna menggunakan rantai penyedia kredensi. Selanjutnya, Anda membuat tugas migrasi data di AWS SCT, lalu menentukan IAM peran untuk diambil oleh agen ekstraksi data, dan memulai tugas. AWS Security Token Service (AWS STS) menghasilkan kredensial sementara untuk mengakses Amazon S3. Agen ekstraksi data menggunakan kredensial ini untuk mengunggah data ke Amazon S3.

Kemudian, AWS SCT berikan Amazon Redshift dengan peran tersebut. IAM Pada gilirannya, Amazon Redshift mendapatkan kredensial sementara baru dari untuk mengakses AWS STS Amazon S3. Amazon Redshift menggunakan kredensial ini untuk menyalin data dari Amazon S3 ke tabel Amazon Redshift Anda.

Pengaturan keamanan

Agen AWS Schema Conversion Tool dan ekstraksi dapat berkomunikasi melalui Secure Sockets Layer (SSL). Untuk mengaktifkan SSL, siapkan toko kepercayaan dan toko kunci.

Untuk mengatur komunikasi yang aman dengan agen ekstraksi Anda

1. Mulai AWS Schema Conversion Tool.
2. Buka menu Pengaturan, lalu pilih Pengaturan global. Kotak dialog Pengaturan global muncul.
3. Pilih Keamanan.
4. Pilih Hasilkan kepercayaan dan toko kunci, atau pilih Pilih toko kepercayaan yang ada.

Jika Anda memilih Hasilkan kepercayaan dan penyimpanan kunci, Anda kemudian menentukan nama dan kata sandi untuk kepercayaan dan penyimpanan kunci, dan jalur ke lokasi untuk file yang dihasilkan. Anda menggunakan file-file ini di langkah selanjutnya.

Jika Anda memilih Pilih toko kepercayaan yang ada, Anda kemudian menentukan kata sandi dan nama file untuk kepercayaan dan penyimpanan kunci. Anda menggunakan file-file ini di langkah selanjutnya.

5. Setelah Anda menentukan toko kepercayaan dan toko kunci, pilih OK untuk menutup kotak dialog Pengaturan global.

Mengkonfigurasi lingkungan untuk agen ekstraksi data

Anda dapat menginstal beberapa agen ekstraksi data pada satu host. Namun, kami menyarankan Anda menjalankan satu agen ekstraksi data pada satu host.

Untuk menjalankan agen ekstraksi data Anda, pastikan Anda menggunakan host dengan setidaknya empat vCPUs dan 32 GB memori. Juga, atur memori minimum yang tersedia AWS SCT untuk setidaknya empat GB. Untuk informasi selengkapnya, lihat [Mengkonfigurasi memori tambahan](#).

Konfigurasi optimal dan jumlah host agen tergantung pada situasi spesifik setiap pelanggan. Pastikan Anda mempertimbangkan faktor-faktor seperti jumlah data yang akan dimigrasi, bandwidth jaringan, waktu untuk mengekstrak data, dan sebagainya. Anda dapat melakukan bukti konsep (PoC) terlebih dahulu, dan kemudian mengkonfigurasi agen ekstraksi data dan host Anda sesuai dengan hasil PoC ini.

Instalasi agen ekstraksi

Kami menyarankan Anda menginstal beberapa agen ekstraksi pada komputer individu, terpisah dari komputer yang menjalankan AWS Schema Conversion Tool.

Agen ekstraksi saat ini didukung pada sistem operasi berikut:

- Microsoft Windows
- Perusahaan Topi Merah Linux (RHEL) 6.0
- Ubuntu Linux (versi 14.04 dan lebih tinggi)

Gunakan prosedur berikut untuk menginstal agen ekstraksi. Ulangi prosedur ini untuk setiap komputer tempat Anda ingin menginstal agen ekstraksi.

Untuk memasang agen ekstraksi

1. Jika Anda belum mengunduh file AWS SCT penginstal, ikuti instruksi di [Instalasi dan Konfigurasi AWS Schema Conversion Tool](#) untuk mengunduhnya. File.zip yang berisi file AWS SCT installer juga berisi file installer agen ekstraksi.
2. Unduh dan instal versi terbaru Amazon Corretto 11. Untuk informasi selengkapnya, lihat [Unduhan untuk Amazon Corretto 11 di Panduan Pengguna Amazon Corretto 11](#).
3. Temukan file penginstal untuk agen ekstraksi Anda di subfolder bernama agen. Untuk setiap sistem operasi komputer, file yang benar untuk menginstal agen ekstraksi ditunjukkan sebagai berikut.

Sistem operasi	Nama file
Microsoft Windows	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .msi
RHEL	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .x86_64.rpm
Ubuntu Linux	aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .deb

4. Instal agen ekstraksi pada komputer terpisah dengan menyalin file penginstal ke komputer baru.
5. Jalankan file installer. Gunakan instruksi untuk sistem operasi Anda, yang ditunjukkan berikut.

Sistem operasi	Petunjuk pemasangan
Microsoft Windows	Klik dua kali file untuk menjalankan penginstal.

Sistem operasi	Petunjuk pemasangan
RHEL	<p>Jalankan perintah berikut di folder yang Anda unduh atau pindahkan file.</p> <pre>sudo rpm -ivh aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .x86_64.rpm sudo ./sct-extractor-setup.sh --config</pre>
Ubuntu Linux	<p>Jalankan perintah berikut di folder yang Anda unduh atau pindahkan file.</p> <pre>sudo dpkg -i aws-schema-conversion-tool-extractor-2.0.1. <i>build-number</i> .deb sudo ./sct-extractor-setup.sh --config</pre>

- Pilih Berikutnya, terima perjanjian lisensi, dan pilih Berikutnya.
- Masukkan jalur untuk menginstal agen ekstraksi AWS SCT data, dan pilih Berikutnya.
- Pilih Instal untuk menginstal agen ekstraksi data Anda.

AWS SCT menginstal agen ekstraksi data Anda. Untuk menyelesaikan instalasi, konfigurasi agen ekstraksi data Anda. AWS SCT secara otomatis meluncurkan program pengaturan konfigurasi. Untuk informasi selengkapnya, lihat [Mengkonfigurasi agen ekstraksi](#).

- Pilih Selesai untuk menutup wizard instalasi setelah Anda mengonfigurasi agen ekstraksi data Anda.

Mengkonfigurasi agen ekstraksi

Gunakan prosedur berikut untuk mengkonfigurasi agen ekstraksi. Ulangi prosedur ini di setiap komputer yang memiliki agen ekstraksi terpasang.

Untuk mengkonfigurasi agen ekstraksi Anda

- Luncurkan program pengaturan konfigurasi:
 - Di Windows, AWS SCT meluncurkan program pengaturan konfigurasi secara otomatis selama instalasi agen ekstraksi data.

Sesuai kebutuhan, Anda dapat meluncurkan program pengaturan secara manual. Untuk melakukannya, jalankan `ConfigAgent.bat` file di Windows. Anda dapat menemukan file ini di folder tempat Anda menginstal agen.

- Di RHEL dan Ubuntu, jalankan `sct-extractor-setup.sh` file dari lokasi tempat Anda menginstal agen.

Program pengaturan meminta Anda untuk informasi. Untuk setiap prompt, nilai default muncul.

2. Terima nilai default pada setiap prompt, atau masukkan nilai baru.

Tentukan informasi berikut:

- Untuk port Listening, masukkan nomor port yang didengarkan agen.
- Untuk Tambahkan vendor sumber, masukkan ya, lalu masukkan platform gudang data sumber Anda.
- Untuk JDBCdriver, masukkan lokasi tempat Anda menginstal JDBC driver.
- Untuk folder Kerja, masukkan jalur tempat agen ekstraksi AWS SCT data akan menyimpan data yang diekstraksi. Folder kerja dapat berada di komputer yang berbeda dari agen, dan satu folder kerja dapat dibagikan oleh beberapa agen di komputer yang berbeda.
- Untuk Aktifkan SSL komunikasi, masukkan ya.
- Untuk Key Store, masukkan lokasi file penyimpanan kunci.
- Untuk kata sandi Key Store, masukkan kata sandi untuk toko kunci.
- Untuk Aktifkan SSL otentikasi klien, masukkan ya.
- Untuk Trust store, masukkan lokasi file trust store.
- Untuk kata sandi toko Trust, masukkan kata sandi untuk toko kepercayaan.

Program pengaturan memperbarui file pengaturan untuk agen ekstraksi. File pengaturan diberi `namasettings.properties`, dan terletak di mana Anda menginstal agen ekstraksi.

Berikut ini adalah contoh file pengaturan.

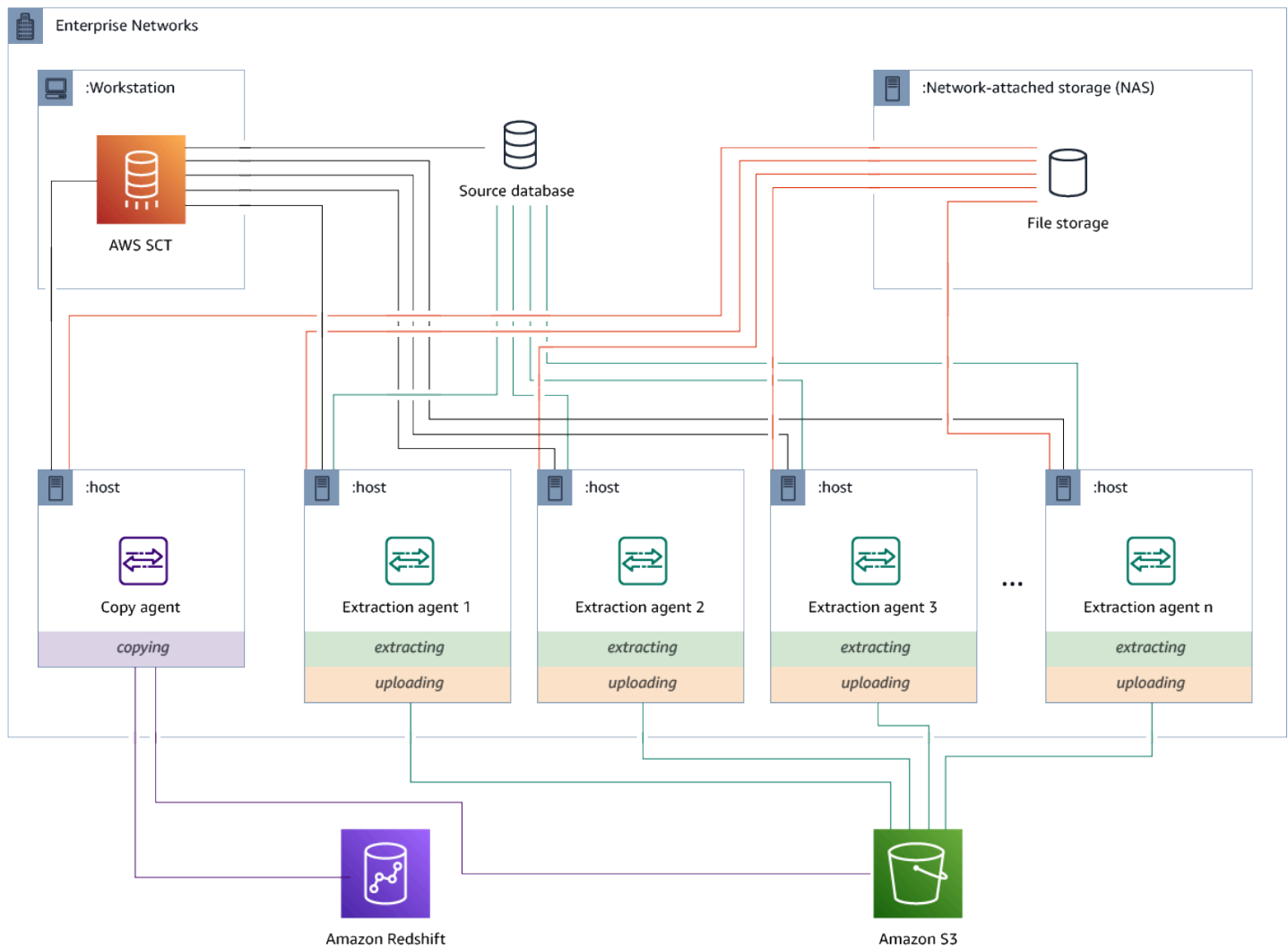
```
$ cat settings.properties  
#extractor.start.fetch.size=20000  
#extractor.out.file.size=10485760  
#extractor.source.connection.pool.size=20  
#extractor.source.connection.pool.min.evictable.idle.time.millis=30000
```

```
#extractor.extracting.thread.pool.size=10
vendor=TERADATA
driver.jars=/usr/share/lib/jdbc/terajdbc4.jar
port=8192
redshift.driver.jars=/usr/share/lib/jdbc/RedshiftJDBC42-1.2.43.1067.jar
working.folder=/data/sct
extractor.private.folder=/home/ubuntu
ssl.option=OFF
```

Untuk mengubah pengaturan konfigurasi, Anda dapat mengedit `settings.properties` file menggunakan editor teks atau menjalankan konfigurasi agen lagi.

Memasang dan mengonfigurasi agen ekstraksi dengan agen penyalinan khusus

Anda dapat menginstal agen ekstraksi dalam konfigurasi yang memiliki penyimpanan bersama dan agen penyalinan khusus. Diagram berikut menggambarkan skenario ini.



Konfigurasi itu dapat berguna ketika server basis data sumber mendukung hingga 120 koneksi, dan jaringan Anda memiliki penyimpanan yang cukup. Gunakan prosedur berikut untuk mengonfigurasi agen ekstraksi yang memiliki agen penyalinan khusus.

Untuk menginstal dan mengonfigurasi agen ekstraksi dan agen penyalinan khusus

1. Pastikan bahwa direktori kerja semua agen ekstraksi menggunakan folder yang sama pada penyimpanan bersama.
2. Instal agen ekstraktor dengan mengikuti langkah-langkahnya. [Instalasi agen ekstraksi](#)
3. Konfigurasi agen ekstraksi dengan mengikuti langkah-langkah [Mengkonfigurasi agen ekstraksi](#), tetapi tentukan hanya JDBC driver sumber.
4. Konfigurasi agen penyalinan khusus dengan mengikuti langkah-langkahnya [Mengkonfigurasi agen ekstraksi](#), tetapi tentukan hanya driver Amazon JDBC Redshift.

Memulai agen ekstraksi

Gunakan prosedur berikut untuk memulai agen ekstraksi. Ulangi prosedur ini di setiap komputer yang memiliki agen ekstraksi terpasang.

Agan ekstraksi bertindak sebagai pendengar. Ketika Anda memulai agen dengan prosedur ini, agen mulai mendengarkan instruksi. Anda mengirim instruksi agen untuk mengekstrak data dari gudang data Anda di bagian selanjutnya.

Untuk memulai agen ekstraksi Anda

- Pada komputer yang memiliki agen ekstraksi diinstal, jalankan perintah yang tercantum berikut untuk sistem operasi Anda.

Sistem operasi	Mulai perintah
Microsoft Windows	Klik dua kali file <code>StartAgent.bat</code> batch.
RHEL	Jalankan perintah berikut di jalur ke folder yang Anda instal agen: <code>sudo initctl <i>start</i> sct-extractor</code>
Ubuntu Linux	Jalankan perintah berikut di jalur ke folder yang Anda instal agen. Gunakan perintah yang sesuai untuk versi Ubuntu Anda. Ubuntu 14.04: <code>sudo initctl <i>start</i> sct-extractor</code> Ubuntu 15.04 dan lebih tinggi: <code>sudo systemctl <i>start</i> sct-extractor</code>

Untuk memeriksa status agen, jalankan perintah yang sama tetapi ganti `start` dengan `status`.

Untuk menghentikan agen, jalankan perintah yang sama tetapi ganti `start` dengan `stop`.

Mendaftarkan agen ekstraksi dengan AWS Schema Conversion Tool

Anda mengelola agen ekstraksi Anda dengan menggunakan AWS SCT. Agen ekstraksi bertindak sebagai pendengar. Ketika mereka menerima instruksi dari AWS SCT, mereka mengekstrak data dari gudang data Anda.

Gunakan prosedur berikut untuk mendaftarkan agen ekstraksi dengan AWS SCT proyek Anda.

Untuk mendaftarkan agen ekstraksi

1. Mulai AWS Schema Conversion Tool, dan buka proyek.
2. Buka menu Tampilan, lalu pilih tampilan Migrasi Data (lainnya). Tab Agen muncul. Jika Anda memiliki agen terdaftar sebelumnya AWS SCT, tampilkan mereka dalam kotak di bagian atas tab.
3. Pilih Pendaftaran.

Setelah Anda mendaftarkan agen dengan AWS SCT proyek, Anda tidak dapat mendaftarkan agen yang sama dengan proyek yang berbeda. Jika Anda tidak lagi menggunakan agen dalam sebuah AWS SCT proyek, Anda dapat membatalkan pendaftarannya. Anda kemudian dapat mendaftarkannya dengan proyek yang berbeda.

4. Pilih agen data Redshift, lalu pilih OK.
5. Masukkan informasi Anda pada tab Koneksi pada kotak dialog:
 - a. Untuk Deskripsi, masukkan deskripsi agen.
 - b. Untuk Nama Host, masukkan nama host atau alamat IP komputer agen.
 - c. Untuk Port, masukkan nomor port yang didengarkan agen.
 - d. Pilih Daftar untuk mendaftarkan agen dengan AWS SCT proyek Anda.
6. Ulangi langkah sebelumnya untuk mendaftarkan beberapa agen dengan AWS SCT proyek Anda.

Menyembunyikan dan memulihkan informasi untuk agen AWS SCT

AWS SCT Agen mengenkripsi sejumlah besar informasi, misalnya kata sandi ke toko kepercayaan kunci pengguna, akun basis data, informasi AWS akun, dan item serupa. Ia melakukannya dengan

menggunakan file khusus yang disebut `seed.dat`. Secara default, agen membuat file ini di folder kerja pengguna yang pertama kali mengkonfigurasi agen.

Karena pengguna yang berbeda dapat mengkonfigurasi dan menjalankan agen, jalur ke `seed.dat` disimpan dalam `{extractor.private.folder}` parameter `settings.properties` file. Ketika agen mulai, ia dapat menggunakan jalur ini untuk menemukan `seed.dat` file untuk mengakses informasi penyimpanan `key-trust` untuk database tempat ia bertindak.

Anda mungkin perlu memulihkan kata sandi yang disimpan agen dalam kasus ini:

- Jika pengguna kehilangan `seed.dat` file dan lokasi serta port AWS SCT agen tidak berubah.
- Jika pengguna kehilangan `seed.dat` file dan lokasi dan port AWS SCT agen telah berubah. Dalam hal ini, perubahan biasanya terjadi karena agen dimigrasikan ke host atau port lain dan informasi dalam `seed.dat` file tidak lagi valid.

Dalam kasus ini, jika agen dimulai tanpa SSL, itu dimulai dan kemudian mengakses penyimpanan agen yang dibuat sebelumnya. Kemudian pergi ke keadaan Menunggu pemulihan.

Namun, dalam kasus ini, jika agen dimulai, SSL Anda tidak dapat memulai ulang. Ini karena agen tidak dapat mendekripsi kata sandi ke sertifikat yang disimpan dalam file `settings.properties`. Dalam jenis startup ini, agen gagal memulai. Kesalahan yang mirip dengan berikut ini ditulis dalam log: "Agen tidak dapat memulai dengan SSL mode diaktifkan. Harap konfigurasi ulang agen. Alasan: Kata sandi untuk keystore tidak benar."

Untuk memperbaikinya, buat agen baru dan konfigurasi agen untuk menggunakan kata sandi yang ada untuk mengakses SSL sertifikat. Untuk melakukannya, gunakan prosedur berikut.

Setelah Anda melakukan prosedur ini, agen harus menjalankan dan pergi ke status Menunggu pemulihan. AWS SCT secara otomatis mengirimkan kata sandi yang diperlukan ke agen dalam keadaan Menunggu pemulihan. Ketika agen memiliki kata sandi, ia memulai ulang tugas apa pun. Tidak ada tindakan pengguna lebih lanjut yang diperlukan di AWS SCT samping.

Untuk mengkonfigurasi ulang agen dan mengembalikan kata sandi untuk mengakses sertifikat SSL

1. Instal AWS SCT agen baru dan jalankan konfigurasi.
2. Ubah `agent.name` properti dalam `instance.properties` file menjadi nama agen tempat penyimpanan dibuat, agar agen baru bekerja dengan penyimpanan agen yang ada.

`instance.properties`File disimpan dalam folder pribadi agen, yang dinamai menggunakan konvensi berikut:`{output.folder}\dmt\{hostName}_{portNumber}\`.

3. Ubah nama `{output.folder}` ke folder keluaran agen sebelumnya.

Pada titik ini, AWS SCT masih mencoba untuk mengakses extractor lama di host lama dan port. Akibatnya, ekstraktor yang tidak dapat diakses mendapatkan status. FAILED Anda kemudian dapat mengubah host dan port.

4. Ubah host, port, atau keduanya untuk agen lama dengan menggunakan perintah Modify untuk mengarahkan aliran permintaan ke agen baru.

Ketika AWS SCT dapat melakukan ping ke agen baru, AWS SCT menerima status Menunggu pemulihan dari agen. AWS SCT kemudian secara otomatis memulihkan kata sandi untuk agen.

Setiap agen yang bekerja dengan penyimpanan agen memperbarui file khusus yang disebut `storage.lock` terletak di `{output.folder}\{agentName}\storage\`. File ini berisi ID jaringan agen dan waktu penyimpanan terkunci. Ketika agen bekerja dengan penyimpanan agen, ia memperbarui `storage.lock` file dan memperpanjang sewa penyimpanan dengan 10 menit setiap 5 menit. Tidak ada contoh lain yang dapat bekerja dengan penyimpanan agen ini sebelum masa sewa berakhir.

Membuat aturan migrasi data di AWS SCT

Sebelum Anda mengekstrak data Anda dengan AWS Schema Conversion Tool, Anda dapat mengatur filter yang mengurangi jumlah data yang Anda ekstrak. Anda dapat membuat aturan migrasi data dengan menggunakan WHERE klausa untuk mengurangi data yang Anda ekstrak. Misalnya, Anda dapat menulis WHERE klausa yang memilih data dari satu tabel.

Anda dapat membuat aturan migrasi data dan menyimpan filter sebagai bagian dari proyek Anda. Dengan proyek Anda terbuka, gunakan prosedur berikut untuk membuat aturan migrasi data.

Untuk membuat aturan migrasi data

1. Buka menu Tampilan, lalu pilih tampilan Migrasi Data (lainnya).
2. Pilih Aturan migrasi data, lalu pilih Tambahkan aturan baru.
3. Konfigurasi aturan migrasi data Anda:
 - a. Untuk Nama, masukkan nama untuk aturan migrasi data Anda.

- b. Untuk Di mana nama skema seperti, masukkan filter untuk diterapkan ke skema. Dalam filter ini, WHERE klausa dievaluasi dengan menggunakan klausa. LIKE Untuk memilih satu skema, masukkan nama skema yang tepat. Untuk memilih beberapa skema, gunakan karakter “%” sebagai wildcard untuk mencocokkan sejumlah karakter dalam nama skema.
 - c. Untuk nama tabel seperti, masukkan filter untuk diterapkan ke tabel. Dalam filter ini, WHERE klausa dievaluasi dengan menggunakan klausa. LIKE Untuk memilih satu tabel, masukkan nama yang tepat. Untuk memilih beberapa tabel, gunakan karakter “%” sebagai wildcard untuk mencocokkan sejumlah karakter dalam nama tabel.
 - d. Untuk klausa Where, masukkan WHERE klausa untuk memfilter data.
4. Setelah mengonfigurasi filter, pilih Simpan untuk menyimpan filter, atau Batalkan untuk membatalkan perubahan.
 5. Setelah Anda selesai menambahkan, mengedit, dan menghapus filter, pilih Simpan semua untuk menyimpan semua perubahan Anda.

Untuk mematikan filter tanpa menghapusnya, gunakan ikon sakelar. Untuk menduplikasi filter yang ada, gunakan ikon salin. Untuk menghapus filter yang ada, gunakan ikon hapus. Untuk menyimpan perubahan apa pun yang Anda buat pada filter, pilih Simpan semua.

Mengubah pengaturan ekstraktor dan salin dari pengaturan proyek

Dari jendela Pengaturan proyek AWS SCT, Anda dapat memilih pengaturan untuk agen ekstraksi data dan perintah Amazon RedshiftCOPY.

Untuk memilih pengaturan ini, pilih Pengaturan, Pengaturan proyek, lalu pilih Migrasi data. Di sini, Anda dapat mengedit pengaturan Ekstraksi, pengaturan Amazon S3, dan pengaturan Salin.

Gunakan petunjuk dalam tabel berikut untuk memberikan informasi untuk pengaturan Ekstraksi.

Untuk parameter ini	Lakukan hal berikut
Format kompresi	Tentukan format kompresi file input. Pilih salah satu opsi berikut: GZIP,, BZIP2ZSTD, atau Tidak ada kompresi.
Karakter pembatas	Tentukan ASCII karakter yang memisahkan bidang dalam file input. Karakter nonprinting tidak didukung.

Untuk parameter ini	Lakukan hal berikut
NULL nilai sebagai string	Aktifkan opsi ini jika data Anda menyertakan terminator null. Jika opsi ini dimatikan, COPY perintah Amazon Redshift memperlakukan null sebagai akhir dari catatan dan mengakhiri proses pemuatan.
Strategi penyortiran	Gunakan penyortiran untuk memulai kembali ekstraksi dari titik kegagalan. Pilih salah satu strategi penyortiran berikut: Gunakan penyortiran setelah kegagalan pertama (disarankan), Gunakan penyortiran jika memungkinkan, atau Jangan pernah menggunakan penyortiran. Untuk informasi selengkapnya, lihat the section called “Menyortir data” .
Skema suhu sumber	Masukkan nama skema dalam database sumber, di mana agen ekstraksi dapat membuat objek sementara.
Ukuran file keluar (dalam MB)	Masukkan ukuran, dalam MB, file yang diunggah ke Amazon S3.
Ukuran file Snowball out (dalam MB)	Masukkan ukuran, dalam MB, file yang diunggah ke AWS Snowball. File bisa berukuran 1—1.000 MB.
Gunakan partisi otomatis. Untuk Greenplum dan Netezza, masukkan ukuran minimal tabel yang didukung (dalam megabyte)	Aktifkan opsi ini untuk menggunakan partisi tabel, lalu masukkan ukuran tabel untuk dipartisi untuk database sumber Greenplum dan Netezza. Untuk migrasi Oracle ke Amazon Redshift, Anda dapat membiarkan bidang ini kosong AWS SCT karena membuat subtask untuk semua tabel yang dipartisi.
Ekstrak LOBs	Aktifkan opsi ini untuk mengekstrak objek besar (LOBs) dari database sumber Anda. LOB termasuk BLOBs, CLOBs, NCLOBs, XML file, dan sebagainya. Untuk setiap LOB, agen AWS SCT ekstraksi membuat file data.
Folder ember Amazon S3 LOBs	Masukkan lokasi untuk agen AWS SCT ekstraksi untuk disimpan LOBs.
Terapkan RTRIM ke kolom string	Aktifkan opsi ini untuk memangkas sekumpulan karakter tertentu dari akhir string yang diekstraksi.

Untuk parameter ini	Lakukan hal berikut
Simpan file secara lokal setelah diunggah ke Amazon S3	Aktifkan opsi ini untuk menyimpan file di komputer lokal Anda setelah agen ekstraksi data mengunggahnya ke Amazon S3.

Gunakan petunjuk dalam tabel berikut untuk memberikan informasi setelah Amazon S3.

Untuk parameter ini	Lakukan hal berikut
Gunakan proxy	Aktifkan opsi ini untuk menggunakan server proxy untuk mengunggah data ke Amazon S3. Kemudian pilih protokol transfer data, masukkan nama host, port, nama pengguna, dan kata sandi.
Jenis titik akhir	Pilih FIPS untuk menggunakan titik akhir Standar Pemrosesan Informasi Federal (FIPS). Pilih VPCE untuk menggunakan titik akhir virtual private cloud (VPC). Kemudian untuk VPC endpoint, masukkan Domain Name System (DNS) dari VPC endpoint Anda.
Simpan file di Amazon S3 setelah menyalin ke Amazon Redshift	Aktifkan opsi ini untuk menyimpan file yang diekstrak di Amazon S3 setelah menyalin file-file ini ke Amazon Redshift.

Gunakan petunjuk dalam tabel berikut untuk memberikan informasi untuk pengaturan Salin.

Untuk parameter ini	Lakukan hal berikut
Jumlah kesalahan maksimum	Masukkan jumlah kesalahan pemuatan. Setelah operasi mencapai batas ini, agen ekstraksi AWS SCT data mengakhiri proses pemuatan data. Nilai default adalah 0, yang berarti bahwa agen ekstraksi AWS SCT data melanjutkan beban data terlepas dari kegagalan.

Untuk parameter ini	Lakukan hal berikut
Ganti karakter UTF -8 yang tidak valid	Aktifkan opsi ini untuk mengganti karakter UTF -8 yang tidak valid dengan karakter yang ditentukan dan lanjutkan operasi pemuatan data.
Gunakan blank sebagai nilai null	Aktifkan opsi ini untuk memuat bidang kosong yang terdiri dari karakter spasi putih sebagai nol.
Gunakan kosong sebagai nilai null	Aktifkan opsi ini untuk memuat kosong CHAR dan VARCHAR bidang sebagai null.
Memotong kolom	Aktifkan opsi ini untuk memotong data dalam kolom agar sesuai dengan spesifikasi tipe data.
Kompresi otomatis	Aktifkan opsi ini untuk menerapkan pengkodean kompresi selama operasi penyalinan.
Penyegaran statistik otomatis	Aktifkan opsi ini untuk menyegarkan statistik di akhir operasi penyalinan.
Periksa file sebelum memuat	Aktifkan opsi ini untuk memvalidasi file data sebelum memuatnya ke Amazon Redshift.

Menyortir data sebelum bermigrasi menggunakan AWS SCT

Menyortir data Anda sebelum migrasi dengan AWS SCT memberikan beberapa manfaat. Jika Anda mengurutkan data terlebih dahulu, AWS SCT dapat memulai ulang agen ekstraksi pada titik terakhir yang disimpan setelah kegagalan. Selain itu, jika Anda memigrasikan data ke Amazon Redshift dan mengurutkan data terlebih dahulu AWS SCT, dapat menyisipkan data ke Amazon Redshift lebih cepat.

Manfaat ini berkaitan dengan cara AWS SCT membuat kueri ekstraksi data. Dalam beberapa kasus, AWS SCT menggunakan fungsi `DENSE _ RANK` analitik dalam kueri ini. Namun, `DENSE _ RANK` dapat menggunakan banyak waktu dan sumber daya server untuk mengurutkan kumpulan data yang dihasilkan dari ekstraksi, jadi jika AWS SCT dapat bekerja tanpanya, itu berhasil.

Untuk mengurutkan data sebelum bermigrasi menggunakan AWS SCT

1. Buka AWS SCT proyek.
2. Buka menu konteks (klik kanan) untuk objek, lalu pilih Buat tugas Lokal.
3. Pilih tab Advanced, dan untuk strategi Sorting, pilih opsi:
 - Jangan pernah menggunakan penyortiran — Agen ekstraksi tidak menggunakan fungsi RANK analitik DENSE _ dan memulai ulang dari awal jika terjadi kegagalan.
 - Gunakan penyortiran jika memungkinkan — Agen ekstraksi menggunakan DENSE _ RANK jika tabel memiliki kunci primer atau kendala unik.
 - Gunakan penyortiran setelah gagal pertama (disarankan) — Agen ekstraksi pertama-tama mencoba mendapatkan data tanpa menggunakan DENSE _RANK. Jika upaya pertama gagal, agen ekstraksi membangun kembali kueri menggunakan DENSE _ RANK dan mempertahankan lokasinya jika terjadi kegagalan.

Create Local task

General | **Advanced** | Source server | AWS S3 settings | Source SSL settings

Extraction settings

Delimiter character: |

Compression format: GZIP

NULL value as a string

Sorting strategy: Use sorting after first fail (recommen...)

Source temp schema:

Out file size (in MB): 10

Apply RTRIM to string columns

Keep files locally after upload to AWS S3

Use subtasks auto-balancing between agents

Freezing interval: 10

Copy settings

Maximum error count: 0

Replace invalid UTF-8 character: ?

Use blank as null value
BLANKSASNULL: This option loads blank fields, which consist of only white space characters, as NULL. The default behavior, without this option, is to load the space characters as is.

Use empty as null value
EMPTYASNULL: This option indicates that Amazon Redshift should load empty CHAR and VARCHAR fields as NULL.

Truncate columns
TRUNCATECOLUMNS: This option truncates data in columns to the appropriate number of characters so that it fits the column specification. This option applies only to columns with a VARCHAR or CHAR data type, and rows 4 MB or less in size.

Automatic compression
COMPUPDATE: This option controls whether compression encodings are automatically

Test Task | Cancel | Create

4. Tetapkan parameter tambahan seperti yang dijelaskan berikut, lalu pilih Buat untuk membuat tugas ekstraksi data Anda.

Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data

Gunakan prosedur berikut untuk membuat, menjalankan, dan memantau tugas ekstraksi data.

Untuk menetapkan tugas ke agen dan memigrasikan data

1. Dalam AWS Schema Conversion Tool, setelah Anda mengonversi skema Anda, pilih satu atau beberapa tabel dari panel kiri proyek Anda.

Anda dapat memilih semua tabel, tetapi kami sarankan untuk tidak melakukannya karena alasan kinerja. Kami menyarankan Anda membuat beberapa tugas untuk beberapa tabel berdasarkan ukuran tabel di gudang data Anda.

2. Buka menu konteks (klik kanan) untuk setiap tabel, lalu pilih Buat tugas. Kotak dialog Create Local task terbuka.
3. Untuk nama Tugas, masukkan nama untuk tugas tersebut.
4. Untuk mode Migrasi, pilih salah satu dari berikut ini:
 - Ekstrak saja - Ekstrak data Anda, dan simpan data ke folder kerja lokal Anda.
 - Ekstrak dan unggah - Ekstrak data Anda, dan unggah data Anda ke Amazon S3.
 - Ekstrak, unggah, dan salin — Ekstrak data Anda, unggah data Anda ke Amazon S3, dan salin ke gudang data Amazon Redshift Anda.
5. Untuk jenis Enkripsi, pilih salah satu dari berikut ini:
 - NONE— Matikan enkripsi data untuk seluruh proses migrasi data.
 - CSE_SK — Gunakan enkripsi sisi klien dengan kunci simetris untuk memigrasikan data. AWS SCT secara otomatis menghasilkan kunci enkripsi dan mengirimkannya ke agen ekstraksi data menggunakan Secure Sockets Layer (SSL). AWS SCT tidak mengenkripsi objek besar (LOBs) selama migrasi data.
6. Pilih Ekstrak LOBs untuk mengekstrak benda besar. Jika Anda tidak perlu mengekstrak benda besar, Anda dapat menghapus kotak centang. Melakukan hal ini mengurangi jumlah data yang Anda ekstrak.
7. Untuk melihat informasi terperinci tentang tugas, pilih Aktifkan pencatatan tugas. Anda dapat menggunakan log tugas untuk men-debug masalah.

Jika Anda mengaktifkan pencatatan tugas, pilih tingkat detail yang ingin Anda lihat. Levelnya adalah sebagai berikut, dengan setiap level termasuk semua pesan dari level sebelumnya:

- ERROR— Jumlah detail terkecil.
- WARNING
- INFO

- DEBUG
 - TRACE— Jumlah detail terbesar.
8. Untuk mengekspor data dari BigQuery, AWS SCT gunakan folder bucket Google Cloud Storage. Dalam folder ini, agen ekstraksi data menyimpan data sumber Anda.

Untuk memasukkan jalur ke folder bucket Google Cloud Storage Anda, pilih Advanced. Untuk folder bucket Google CS, masukkan nama bucket dan nama folder.
 9. Untuk mengambil peran bagi pengguna agen ekstraksi data Anda, pilih pengaturan Amazon S3. Untuk IAMperan, masukkan nama peran yang akan digunakan. Untuk Wilayah, pilih Wilayah AWS untuk peran ini.
 10. Pilih Tugas uji untuk memverifikasi bahwa Anda dapat terhubung ke folder kerja, bucket Amazon S3, dan gudang data Amazon Redshift. Verifikasi tergantung pada mode migrasi yang Anda pilih.
 11. Memilih Membuat untuk membuat tugas.
 12. Ulangi langkah sebelumnya untuk membuat tugas untuk semua data yang ingin Anda migrasi.

Untuk menjalankan dan memantau tugas

1. Untuk Tampilan, pilih tampilan Migrasi Data. Tab Agen muncul.
2. Pilih tab Tugas. Tugas Anda muncul di kisi di bagian atas seperti yang ditunjukkan berikut. Anda dapat melihat status tugas di kisi atas, dan status subtugasnya di kisi bawah.

The screenshot displays the AWS SCT interface with the 'Tasks' tab selected. The main area shows a table of migration tasks with columns for Name, Extract, Upload, and Copy. Each task has a progress bar and a status icon (green checkmark for success, yellow play button for pending, and red X for failure).

Name	Extract	Upload	Copy
+ CUSTOMER	+ 0%		
+ LINEORDER_100K	+ 0%		
+ LINEORDER_150K	+ 0%		
+ LINEORDER_1M	+ 0%		
LocalTask 2	100%	100%	
+ CUSTOMER	100%	100%	
+ LINEORDER_100K	100%	100%	
+ LINEORDER_150K	100%	100%	
LocalTask 3	100%	100%	0%
+ LINEORDER_100K	100%	100%	0%

Below the table, there are control buttons: Download log, All migration Tasks, Resume, Stop, Restart, Reset, Delete, Replace, Refresh all, and Refresh selected. At the bottom, the 'Processing details' tab is active, showing a progress bar for 'Extracting Data' at 100%.

3. Pilih tugas di kisi atas dan perluas. Bergantung pada mode migrasi yang Anda pilih, Anda melihat tugas dibagi menjadi Ekstrak, Unggah, dan Salin.
4. Pilih Mulai untuk tugas untuk memulai tugas itu. Anda dapat memantau status tugas Anda saat mereka bekerja. Subtask berjalan secara paralel. Ekstrak, unggah, dan salin juga berjalan secara paralel.
5. Jika Anda mengaktifkan logging saat mengatur tugas, Anda dapat melihat log:
 - a. Pilih Unduh log. Sebuah pesan muncul dengan nama folder yang berisi file log. Singkirkan pesannya.
 - b. Tautan muncul di tab Detail tugas. Pilih tautan untuk membuka folder yang berisi file log.

Anda dapat menutup AWS SCT, dan agen serta tugas Anda terus berjalan. Anda dapat membuka kembali AWS SCT nanti untuk memeriksa status tugas Anda dan melihat log tugas.

Anda dapat menyimpan tugas ekstraksi data ke disk lokal Anda dan mengembalikannya ke proyek yang sama atau lainnya dengan menggunakan ekspor dan impor. Untuk mengekspor tugas, pastikan Anda memiliki setidaknya satu tugas ekstraksi yang dibuat dalam sebuah proyek. Anda dapat mengimpor tugas ekstraksi tunggal atau semua tugas yang dibuat dalam proyek.

Saat Anda mengekspor tugas ekstraksi, AWS SCT buat `.xml` file terpisah untuk tugas itu. `.xml`File menyimpan informasi metadata tugas tersebut, seperti properti tugas, deskripsi, dan subtugas. `.xml`File tidak berisi informasi tentang pemrosesan tugas ekstraksi. Informasi seperti berikut ini dibuat ulang saat tugas diimpor:

- Kemajuan tugas
- Subtugas dan status panggung
- Distribusi agen ekstraksi berdasarkan subtugas dan tahapan
- Tugas dan subtugas IDs
- Nama tugas

Mengekspor dan mengimpor tugas ekstraksi AWS SCT data

Anda dapat dengan cepat menyimpan tugas yang ada dari satu proyek dan mengembalikannya di proyek lain (atau proyek yang sama) menggunakan AWS SCT ekspor dan impor. Gunakan prosedur berikut untuk mengekspor dan mengimpor tugas ekstraksi data.

Untuk mengekspor dan mengimpor tugas ekstraksi data

1. Untuk Tampilan, pilih tampilan Migrasi Data. Tab Agen muncul.
2. Pilih tab Tugas. Tugas Anda tercantum dalam kisi yang muncul.
3. Pilih tiga titik yang disejajarkan secara vertikal (ikon elipsis) yang terletak di sudut kanan bawah di bawah daftar tugas.
4. Pilih tugas Ekspor dari menu pop-up.
5. Pilih folder tempat Anda AWS SCT ingin menempatkan .xml file ekspor tugas.

AWS SCT membuat file ekspor tugas dengan format nama file *TASK-DESCRIPTION_TASK-ID.xml*.

6. Pilih tiga titik yang disejajarkan secara vertikal (ikon elipsis) di kanan bawah di bawah daftar tugas.
7. Pilih Impor tugas dari menu pop-up.

Anda dapat mengimpor tugas ekstraksi ke proyek yang terhubung ke database sumber, dan proyek memiliki setidaknya satu agen ekstraksi terdaftar aktif.

8. Pilih .xml file untuk tugas ekstraksi yang Anda ekspor.

AWS SCT mendapatkan parameter tugas ekstraksi dari file, membuat tugas, dan menambahkan tugas ke agen ekstraksi.

9. Ulangi langkah-langkah ini untuk mengekspor dan mengimpor tugas ekstraksi data tambahan.

Pada akhir proses ini, ekspor dan impor Anda selesai dan tugas ekstraksi data Anda siap digunakan.

Ekstraksi data menggunakan perangkat AWS Snowball Edge

Proses penggunaan AWS SCT dan AWS Snowball Edge memiliki beberapa langkah. Migrasi melibatkan tugas lokal, di mana AWS SCT menggunakan agen ekstraksi data untuk memindahkan data ke perangkat AWS Snowball Edge, lalu tindakan perantara di mana AWS menyalin data dari perangkat AWS Snowball Edge ke bucket Amazon S3. Proses selesai AWS SCT memuat data dari bucket Amazon S3 ke Amazon Redshift.

Bagian berikut ikhtisar ini memberikan step-by-step panduan untuk masing-masing tugas ini. Prosedur ini mengasumsikan bahwa Anda telah AWS SCT instal dan bahwa Anda telah mengonfigurasi dan mendaftarkan agen ekstraksi data pada mesin khusus.

Lakukan langkah-langkah berikut untuk memigrasikan data dari penyimpanan data lokal ke penyimpanan AWS data menggunakan AWS Snowball Edge.

1. Buat pekerjaan AWS Snowball Edge menggunakan AWS Snowball konsol.
2. Buka kunci perangkat AWS Snowball Edge menggunakan mesin Linux lokal khusus.
3. Buat proyek baru di AWS SCT.
4. Instal dan konfigurasi agen ekstraksi data Anda.
5. Buat dan setel izin untuk bucket Amazon S3 untuk digunakan.
6. Impor AWS Snowball pekerjaan ke AWS SCT proyek Anda.
7. Daftarkan agen ekstraksi data Anda di AWS SCT.
8. Buat tugas lokal di AWS SCT.
9. Jalankan dan pantau tugas migrasi data di AWS SCT.

tep-by-step Prosedur S untuk memigrasi data menggunakan AWS SCT dan AWS Snowball Edge

Bagian berikut memberikan informasi rinci tentang langkah-langkah migrasi.

Langkah 1: Buat pekerjaan AWS Snowball Edge

Buat AWS Snowball pekerjaan dengan mengikuti langkah-langkah yang diuraikan di bagian [Membuat Pekerjaan AWS Snowball Tepi](#) di Panduan Pengembang AWS Snowball Edge.

Langkah 2: Buka kunci perangkat AWS Snowball Edge

Jalankan perintah yang membuka kunci dan memberikan kredensial ke perangkat Snowball Edge dari mesin tempat Anda menginstal agen. AWS DMS Dengan menjalankan perintah ini, Anda dapat yakin bahwa panggilan AWS DMS agen terhubung ke perangkat AWS Snowball Edge. Untuk informasi selengkapnya tentang membuka kunci perangkat AWS Snowball Edge, lihat [Membuka Kunci Snowball](#) Edge.

```
aws s3 ls s3://<bucket-name> --profile <Snowball Edge profile> --endpoint http://<Snowball IP>:8080 --recursive
```

Langkah 3: Buat AWS SCT proyek baru

Selanjutnya, buat AWS SCT proyek baru.

Untuk membuat proyek baru di AWS SCT

1. Mulai AWS Schema Conversion Tool. Pada menu File, pilih Proyek baru. Kotak dialog Proyek baru muncul.
2. Memasukkan nama untuk proyek Anda, yang menyimpan secara lokal di komputer Anda.
3. Memasukkan lokasi untuk file proyek lokal Anda.
4. Pilih OK untuk membuat AWS SCT proyek Anda.
5. Pilih Tambahkan sumber untuk menambahkan database sumber baru ke AWS SCT proyek Anda.
6. Pilih Tambahkan target untuk menambahkan platform target baru di AWS SCT proyek Anda.
7. Pilih skema basis data sumber di panel kiri.
8. Di panel kanan, tentukan platform basis data target untuk skema sumber yang dipilih.
9. Pilih Buat pemetaan. Tombol ini menjadi aktif setelah Anda memilih skema basis data sumber dan platform basis data target.

Langkah 4: Instal dan konfigurasi agen ekstraksi data Anda

AWS SCT menggunakan agen ekstraksi data untuk memigrasikan data ke Amazon Redshift. File.zip yang Anda unduh untuk diinstal AWS SCT, termasuk file penginstal agen ekstraksi. Anda dapat menginstal agen ekstraksi data di Windows, Red Hat Enterprise Linux, atau Ubuntu. Untuk informasi selengkapnya, lihat [Instalasi agen ekstraksi](#).

Untuk mengonfigurasi agen ekstraksi data Anda, masukkan mesin basis data sumber dan target Anda. Selain itu, pastikan Anda mengunduh JDBC driver untuk basis data sumber dan target Anda di komputer tempat Anda menjalankan agen ekstraksi data Anda. Agen ekstraksi data menggunakan driver ini untuk terhubung ke basis data sumber dan target Anda. Untuk informasi selengkapnya, lihat [Menginstal JDBC driver untuk AWS Schema Conversion Tool](#).

Di Windows, penginstal agen ekstraksi data meluncurkan wizard konfigurasi di jendela prompt perintah. Di Linux, jalankan `sct-extractor-setup.sh` file dari lokasi tempat Anda menginstal agen.

Langkah 5: Konfigurasi AWS SCT untuk mengakses bucket Amazon S3

Untuk informasi tentang mengonfigurasi bucket Amazon S3, [lihat ikhtisar Bucket di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#).

Langkah 6: Impor AWS Snowball pekerjaan ke AWS SCT proyek Anda

Untuk menghubungkan AWS SCT proyek Anda dengan perangkat AWS Snowball Edge Anda, impor AWS Snowball pekerjaan Anda.

Untuk mengimpor AWS Snowball pekerjaan Anda

1. Buka menu Pengaturan, lalu pilih Pengaturan global. Kotak dialog Pengaturan global muncul.
2. Pilih profil AWS layanan, lalu pilih Impor pekerjaan.
3. Pilih AWS Snowball pekerjaan Anda.
4. Masukkan AWS Snowball IP Anda. Untuk informasi selengkapnya, lihat [Mengubah Alamat IP Anda](#) di Panduan AWS Snowball Pengguna.
5. Masukkan AWS Snowball Port Anda. Untuk informasi selengkapnya, lihat [Port yang Diperlukan untuk Menggunakan AWS Layanan di Perangkat AWS Snowball Edge](#) di Panduan Pengembang AWS Snowball Edge.
6. Masukkan kunci AWS Snowball akses dan kunci AWS Snowball rahasia Anda. Untuk informasi selengkapnya, lihat [Otorisasi dan Kontrol Akses AWS Snowball di](#) Panduan AWS Snowball Pengguna.
7. Pilih Berlakukan, lalu pilih OKE.

Langkah 7: Daftarkan agen ekstraksi data di AWS SCT

Di bagian ini, Anda mendaftarkan agen ekstraksi data di AWS SCT.

Untuk mendaftarkan agen ekstraksi data

1. Pada menu Tampilan, pilih Tampilan migrasi data (lainnya), lalu pilih Daftar.
2. Untuk Deskripsi, masukkan nama untuk agen ekstraksi data Anda.
3. Untuk nama Host, masukkan alamat IP komputer tempat Anda menjalankan agen ekstraksi data Anda.
4. Untuk Port, masukkan port mendengarkan yang Anda konfigurasi.
5. Pilih Pendaftaran.

Langkah 8: Membuat tugas lokal

Selanjutnya, Anda membuat tugas migrasi. Tugas tersebut mencakup dua subtugas. Satu subtugas memigrasikan data dari database sumber ke alat AWS Snowball Edge. Subtugas lainnya mengambil data yang dimuat alat ke dalam bucket Amazon S3 dan memigrasikannya ke database target.

Untuk membuat tugas migrasi

1. Pada menu Tampilan, lalu pilih Tampilan migrasi data (lainnya).
2. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih objek skema untuk dimigrasi. Buka menu konteks (klik kanan) untuk objek, lalu pilih Buat tugas lokal.
3. Untuk nama Tugas, masukkan nama deskriptif untuk tugas migrasi data Anda.
4. Untuk mode Migrasi, pilih Ekstrak, unggah, dan salin.
5. Pilih pengaturan Amazon S3.
6. Pilih Gunakan Snowball.
7. Masukkan folder dan subfolder di bucket Amazon S3 tempat agen ekstraksi data dapat menyimpan data.
8. Memilih Membuat untuk membuat tugas.

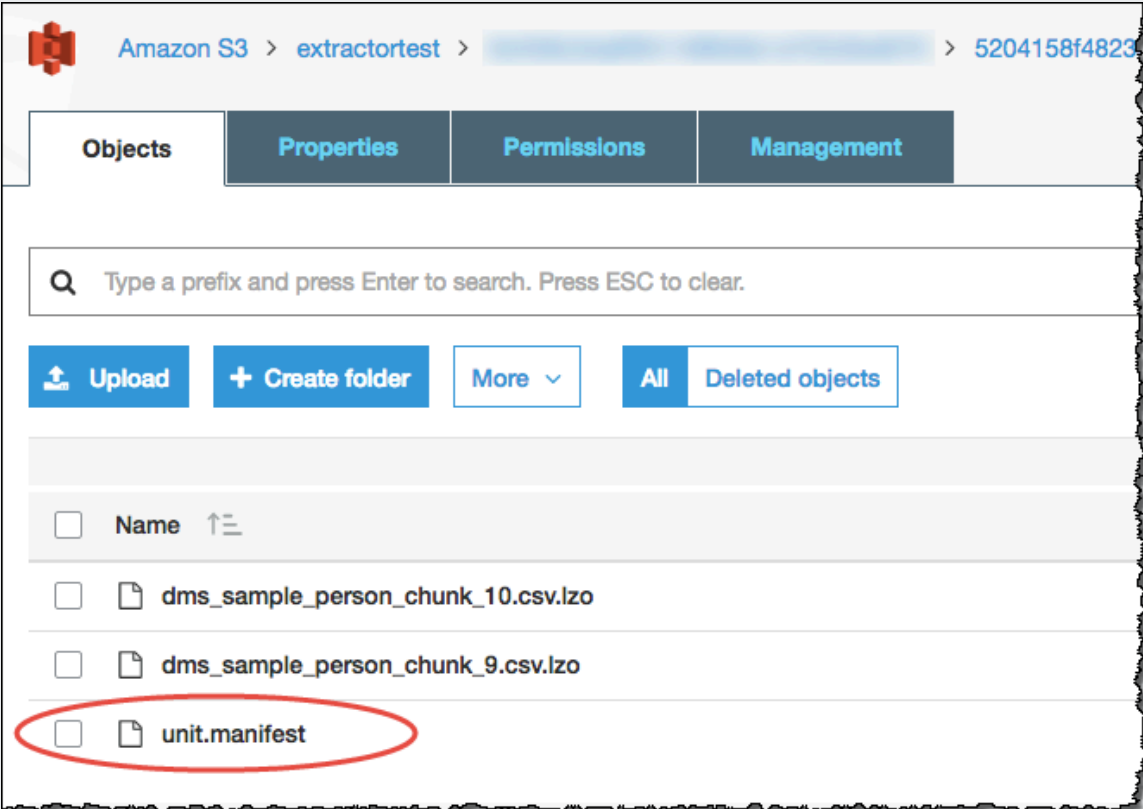
Langkah 9: Menjalankan dan memantau tugas migrasi data di AWS SCT

Untuk memulai tugas migrasi data, pilih Mulai. Pastikan Anda membuat koneksi ke database sumber, bucket Amazon S3, AWS Snowball perangkat, serta koneksi ke database target aktif. AWS

Anda dapat memantau dan mengelola tugas migrasi data dan subtugasnya di tab Tugas. Anda dapat melihat kemajuan migrasi data, serta menjeda atau memulai ulang tugas migrasi data Anda.

Output tugas ekstraksi data

Setelah tugas migrasi Anda selesai, data Anda siap. Gunakan informasi berikut untuk menentukan cara melanjutkan berdasarkan mode migrasi yang Anda pilih dan lokasi data Anda.

Modus migrasi	Lokasi data
Ekstrak, unggah, dan salin	Data sudah ada di gudang data Amazon Redshift Anda. Anda dapat memverifikasi bahwa data ada di sana, dan mulai menggunakannya. Untuk informasi selengkapnya, lihat Menghubungkan ke klaster dari alat dan kode klien .
Ekstrak dan unggah	<p>Agan ekstraksi menyimpan data Anda sebagai file di bucket Amazon S3 Anda. Anda dapat menggunakan COPY perintah Amazon Redshift untuk memuat data Anda ke Amazon Redshift. Untuk informasi selengkapnya, lihat Memuat data dari Amazon S3 di dokumentasi Amazon Redshift.</p> <p>Ada beberapa folder di bucket Amazon S3 Anda, sesuai dengan tugas ekstraksi yang Anda siapkan. Saat memuat data ke Amazon Redshift, tentukan nama file manifes yang dibuat oleh setiap tugas. File manifes muncul di folder tugas di bucket Amazon S3 Anda seperti yang ditunjukkan berikut.</p>  <p>The screenshot shows the Amazon S3 console interface for a bucket named 'extractortest'. The breadcrumb path is 'Amazon S3 > extractortest > 5204158f4823'. There are four tabs: 'Objects', 'Properties', 'Permissions', and 'Management'. Below the tabs is a search bar with the text 'Type a prefix and press Enter to search. Press ESC to clear.' There are four buttons: 'Upload', 'Create folder', 'More', and 'All Deleted objects'. Below the buttons is a list of objects with checkboxes and file icons. The objects are: 'Name', 'dms_sample_person_chunk_10.csv.lzo', 'dms_sample_person_chunk_9.csv.lzo', and 'unit.manifest'. The 'unit.manifest' object is circled in red.</p>

Modus migrasi	Lokasi data
Ekstrak saja	Agan ekstraksi menyimpan data Anda sebagai file di folder kerja Anda. Salin data Anda secara manual ke bucket Amazon S3 Anda, lalu lanjutkan dengan instruksi untuk Ekstrak dan unggah.

Menggunakan partisi virtual dengan AWS Schema Conversion Tool

Anda sering dapat mengelola tabel non-partisi besar dengan membuat subtugas yang membuat partisi virtual dari data tabel menggunakan aturan pemfilteran. Di AWS SCT, Anda dapat membuat partisi virtual untuk data yang dimigrasi. Ada tiga jenis partisi, yang bekerja dengan tipe data tertentu:

- Jenis RANGE partisi bekerja dengan tipe data numerik dan tanggal dan waktu.
- Jenis LIST partisi bekerja dengan tipe data numerik, karakter, dan tanggal dan waktu.
- Jenis DATE AUTO SPLIT partisi bekerja dengan tipe data numerik, tanggal, dan waktu.

AWS SCT memvalidasi nilai yang Anda berikan untuk membuat partisi. Misalnya, jika Anda mencoba mempartisi kolom dengan tipe data NUMERIC tetapi Anda memberikan nilai dari tipe data yang berbeda, akan AWS SCT menimbulkan kesalahan.

Selain itu, jika Anda menggunakan AWS SCT untuk memigrasikan data ke Amazon Redshift, Anda dapat menggunakan partisi asli untuk mengelola migrasi tabel besar. Untuk informasi selengkapnya, lihat [Menggunakan partisi asli](#).

Batas saat membuat partisi virtual

Ini adalah batasan untuk membuat partisi virtual:

- Anda hanya dapat menggunakan partisi virtual hanya untuk tabel yang tidak dipartisi.
- Anda dapat menggunakan partisi virtual hanya dalam tampilan migrasi data.
- Anda tidak dapat menggunakan opsi UNION ALL VIEW dengan partisi virtual.

RANGE jenis partisi

Jenis RANGE partisi partisi data berdasarkan rentang nilai kolom untuk tipe data numerik dan tanggal dan waktu. Jenis partisi ini membuat WHERE klausa, dan Anda memberikan rentang nilai untuk setiap partisi. Untuk menentukan daftar nilai untuk kolom yang dipartisi, gunakan kotak Nilai. Anda dapat memuat informasi nilai dengan menggunakan file.csv.

Jenis RANGE partisi menciptakan partisi default di kedua ujung nilai partisi. Partisi default ini menangkap data apa pun yang kurang dari atau lebih besar dari nilai partisi yang ditentukan.

Misalnya, Anda dapat membuat beberapa partisi berdasarkan rentang nilai yang Anda berikan. Dalam contoh berikut, nilai partisi untuk LO_ ditentukan untuk membuat TAX beberapa partisi.

```
Partition1: WHERE LO_TAX <= 10000.9
Partition2: WHERE LO_TAX > 10000.9 AND LO_TAX <= 15005.5
Partition3: WHERE LO_TAX > 15005.5 AND LO_TAX <= 25005.95
```

Untuk membuat partisi RANGE virtual

1. Buka AWS SCT.
2. Pilih mode tampilan Migrasi Data (lainnya).
3. Pilih tabel tempat Anda ingin mengatur partisi virtual. Buka menu konteks (klik kanan) untuk tabel, dan pilih Tambahkan partisi virtual.
4. Dalam Tambahkan partisi virtual kotak dialog, masukkan informasi sebagai berikut.

Opsi	Tindakan
Jenis partisi	Pilih RANGE. UI kotak dialog berubah tergantung pada jenis yang Anda pilih.
Nama kolom	Pilih kolom yang ingin Anda partisi.
Jenis kolom	Pilih tipe data untuk nilai di kolom.
Nilai-nilai	Tambahkan nilai baru dengan mengetikkan setiap nilai di kotak Nilai Baru, lalu pilih tanda plus untuk menambahkan nilai.
Muat dari file	(Opsional) Masukkan nama file.csv yang berisi nilai partisi.

5. Pilih OKE.

LISTjenis partisi

Jenis LIST partisi partisi partisi data berdasarkan nilai kolom untuk tipe data numerik, karakter, dan tanggal dan waktu. Jenis partisi ini membuat WHERE klausa, dan Anda memberikan nilai untuk setiap partisi. Untuk menentukan daftar nilai untuk kolom yang dipartisi, gunakan kotak Nilai. Anda dapat memuat informasi nilai dengan menggunakan file.csv.

Misalnya, Anda dapat membuat beberapa partisi berdasarkan nilai yang Anda berikan. Dalam contoh berikut, nilai partisi untuk LO_ ditentukan untuk membuat ORDERKEY beberapa partisi.

```
Partition1: WHERE LO_ORDERKEY = 1
Partition2: WHERE LO_ORDERKEY = 2
Partition3: WHERE LO_ORDERKEY = 3
...
PartitionN: WHERE LO_ORDERKEY = USER_VALUE_N
```

Anda juga dapat membuat partisi default untuk nilai-nilai yang tidak termasuk dalam yang ditentukan.

Anda dapat menggunakan jenis LIST partisi untuk memfilter data sumber jika Anda ingin mengecualikan nilai tertentu dari migrasi. Misalnya, anggaplah Anda ingin menghilangkan baris dengan LO_ORDERKEY = 4. Dalam hal ini, jangan sertakan nilai 4 dalam daftar nilai partisi dan pastikan bahwa Sertakan nilai lain tidak dipilih.

Untuk membuat partisi LIST virtual

1. Buka AWS SCT.
2. Pilih mode tampilan Migrasi Data (lainnya).
3. Pilih tabel tempat Anda ingin mengatur partisi virtual. Buka menu konteks (klik kanan) untuk tabel, dan pilih Tambahkan partisi virtual.
4. Dalam Tambahkan partisi virtual kotak dialog, masukkan informasi sebagai berikut.

Opsi	Tindakan
Jenis partisi	Pilih LIST. UI kotak dialog berubah tergantung pada jenis yang Anda pilih.

Opsi	Tindakan
Nama kolom	Pilih kolom yang ingin Anda partisi.
Nilai baru	Ketik nilai di sini untuk menambahkannya ke set nilai partisi.
Sertakan nilai lainnya	Pilih opsi ini untuk membuat partisi default di mana semua nilai yang tidak memenuhi kriteria partisi disimpan.
Muat dari file	(Opsional) Masukkan nama file.csv yang berisi nilai partisi.

5. Pilih OKE.

DATEAUTOSPLIT jenis partisi

Jenis DATE AUTO SPLIT partisi adalah cara otomatis untuk menghasilkan RANGE partisi. Dengan DATA AUTOSPLIT, Anda memberi tahu AWS SCT atribut partisi, di mana untuk memulai dan mengakhiri, dan ukuran rentang antara nilai-nilai. Kemudian AWS SCT menghitung nilai partisi secara otomatis.

DATAAUTOSPLIT mengotomatiskan banyak pekerjaan yang terlibat dengan pembuatan partisi rentang. Tradeoff antara menggunakan teknik ini dan partisi rentang adalah seberapa banyak kontrol yang Anda butuhkan atas batas partisi. Proses pemisahan otomatis selalu menciptakan rentang ukuran (seragam) yang sama. Partisi rentang memungkinkan Anda untuk memvariasikan ukuran setiap rentang sesuai kebutuhan untuk distribusi data tertentu Anda. Misalnya, Anda dapat menggunakan harian, mingguan, dua mingguan, bulanan, dan sebagainya.

```
Partition1: WHERE LO_ORDERDATE >= '1954-10-10' AND LO_ORDERDATE < '1954-10-24'
Partition2: WHERE LO_ORDERDATE >= '1954-10-24' AND LO_ORDERDATE < '1954-11-06'
Partition3: WHERE LO_ORDERDATE >= '1954-11-06' AND LO_ORDERDATE < '1954-11-20'
...
PartitionN: WHERE LO_ORDERDATE >= USER_VALUE_N AND LO_ORDERDATE <= '2017-08-13'
```

Untuk membuat partisi DATE AUTO SPLIT virtual

1. Buka AWS SCT.
2. Pilih mode tampilan Migrasi Data (lainnya).
3. Pilih tabel tempat Anda ingin mengatur partisi virtual. Buka menu konteks (klik kanan) untuk tabel, dan pilih Tambahkan partisi virtual.

4. Dalam Tambahkan partisi virtual kotak dialog, masukkan informasi sebagai berikut.

Opsi	Tindakan
Jenis partisi	Pilih DATEAUTOSPLIT. UI kotak dialog berubah tergantung pada jenis yang Anda pilih.
Nama kolom	Pilih kolom yang ingin Anda partisi.
Tanggal mulai	Ketik tanggal mulai.
Tanggal akhir	Ketik tanggal akhir.
Interval	Masukkan unit interval, dan pilih nilai untuk unit itu.

5. Pilih OKE.

Menggunakan partisi asli

Untuk mempercepat migrasi data, agen ekstraksi data Anda dapat menggunakan partisi asli tabel di server gudang data sumber Anda. AWS SCT mendukung partisi asli untuk migrasi dari Greenplum, Netezza, dan Oracle ke Amazon Redshift.

Misalnya, setelah membuat proyek, Anda dapat mengumpulkan statistik pada skema dan menganalisis ukuran tabel yang dipilih untuk migrasi. Untuk tabel yang melebihi ukuran yang ditentukan, AWS SCT memicu mekanisme partisi asli.

Untuk menggunakan partisi asli

1. Buka AWS SCT, dan pilih Proyek baru untuk File. Kotak dialog Proyek baru muncul.
2. Buat proyek baru, tambahkan server sumber dan target Anda, dan buat aturan pemetaan. Untuk informasi selengkapnya, lihat [Memulai dan mengelola Proyek di AWS SCT](#).
3. Pilih Tampilan, lalu pilih Tampilan utama.
4. Untuk pengaturan Project, pilih tab Migrasi data. Pilih Gunakan partisi otomatis. Untuk database sumber Greenplum dan Netezza, masukkan ukuran minimal tabel yang didukung dalam megabyte (misalnya, 100). AWS SCT secara otomatis membuat subtugas migrasi terpisah untuk setiap partisi asli yang tidak kosong. Untuk migrasi Oracle ke Amazon Redshift, buat subtugas untuk semua tabel AWS SCT yang dipartisi.

5. Di panel kiri yang menampilkan skema dari database sumber Anda, pilih skema. Buka menu konteks (klik kanan) untuk objek, dan pilih Kumpulkan statistik. Untuk migrasi data dari Oracle ke Amazon Redshift, Anda dapat melewati langkah ini.
6. Pilih semua tabel untuk dimigrasi.
7. Daftarkan jumlah agen yang diperlukan. Untuk informasi selengkapnya, lihat [Mendaftarkan agen ekstraksi dengan AWS Schema Conversion Tool](#).
8. Buat tugas ekstraksi data untuk tabel yang dipilih. Untuk informasi selengkapnya, lihat [Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data](#).

Periksa apakah tabel besar dibagi menjadi subtugas, dan setiap subtugas cocok dengan kumpulan data yang menyajikan bagian tabel yang terletak pada satu irisan di gudang data sumber Anda.

9. Mulai dan pantau proses migrasi hingga agen ekstraksi AWS SCT data menyelesaikan migrasi data dari tabel sumber Anda.

Bermigrasi LOBs ke Amazon Redshift

Amazon Redshift tidak mendukung penyimpanan objek biner besar (LOBs). Namun, jika Anda perlu memigrasikan satu atau beberapa LOBs ke Amazon Redshift AWS SCT, dapat melakukan migrasi. Untuk melakukannya, AWS SCT gunakan bucket Amazon S3 untuk menyimpan LOBs dan menulis bucket Amazon S3 ke dalam data migrasi yang disimpan di Amazon Redshift. URL

Untuk bermigrasi LOBs ke Amazon Redshift

1. Buka AWS SCT proyek.
2. Connect ke database sumber dan target. Segarkan metadata dari database target, dan pastikan tabel yang dikonversi ada di sana.
3. Untuk Tindakan, pilih Buat tugas lokal.
4. Untuk mode Migrasi, pilih salah satu dari berikut ini:
 - Ekstrak dan unggah untuk mengekstrak data Anda, dan unggah data Anda ke Amazon S3.
 - Ekstrak, unggah, dan salin untuk mengekstrak data Anda, unggah data Anda ke Amazon S3, dan salin ke gudang data Amazon Redshift Anda.
5. Pilih pengaturan Amazon S3.

- Untuk LOBSfolder bucket Amazon S3, masukkan nama folder di bucket Amazon S3 tempat Anda ingin disimpan. LOBs

Jika Anda menggunakan profil AWS layanan, bidang ini bersifat opsional. AWS SCT dapat menggunakan pengaturan default dari profil Anda. Untuk menggunakan bucket Amazon S3 lainnya, masukkan jalur di sini.

- Aktifkan opsi Gunakan proxy untuk menggunakan server proxy untuk mengunggah data ke Amazon S3. Kemudian pilih protokol transfer data, masukkan nama host, port, nama pengguna, dan kata sandi.
- Untuk tipe Endpoint, pilih FIPS untuk menggunakan titik akhir Federal Information Processing Standard (FIPS). Pilih VPCE untuk menggunakan titik akhir virtual private cloud (VPC). Kemudian untuk VPC endpoint, masukkan Domain Name System (DNS) dari VPC endpoint Anda.
- Aktifkan Simpan file di Amazon S3 setelah menyalin ke Amazon Redshift opsi untuk menyimpan file yang diekstraksi di Amazon S3 setelah menyalin file-file ini ke Amazon Redshift.
- Memilih Membuat untuk membuat tugas.

Praktik terbaik dan pemecahan masalah untuk agen ekstraksi data

Berikut ini adalah beberapa praktik terbaik dan saran pemecahan masalah untuk menggunakan agen ekstraksi.

Masalah	Saran pemecahan masalah
Kinerja lambat	<p>Untuk meningkatkan kinerja, kami merekomendasikan yang berikut:</p> <ul style="list-style-type: none"> Instal beberapa agen. Instal agen di komputer yang dekat dengan gudang data Anda. Jangan menjalankan semua tabel pada satu tugas agen.
Penundaan pertikaian	Hindari terlalu banyak agen yang mengakses gudang data Anda secara bersamaan.
Seorang agen turun sementara	Jika agen down, status setiap tugasnya muncul sebagai gagal dalam AWS SCT. Jika Anda menunggu, dalam beberapa kasus

Masalah	Saran pemecahan masalah
	<p>agen dapat pulih. Dalam hal ini, status tugasnya diperbarui di AWS SCT.</p>
Seorang agen turun secara permanen	<p>Jika komputer yang menjalankan agen mati secara permanen, dan agen tersebut menjalankan tugas, Anda dapat mengganti agen baru untuk melanjutkan tugas. Anda dapat mengganti agen baru hanya jika folder kerja agen asli tidak berada di komputer yang sama dengan agen asli. Untuk mengganti agen baru, lakukan hal berikut:</p> <ul style="list-style-type: none">• Instal agen di komputer baru.• Konfigurasi agen baru dengan pengaturan yang sama, termasuk nomor port dan folder kerja, seperti agen asli.• Mulai agen. Setelah agen dimulai, tugas menemukan agen baru yang tersedia dan terus berjalan pada agen baru.

Mengonversi aplikasi menggunakan SQL AWS SCT

Ketika Anda mengonversi skema database Anda dari satu mesin ke mesin lain, Anda juga perlu memperbarui SQL kode dalam aplikasi Anda untuk berinteraksi dengan mesin database baru, bukan yang lama. Anda dapat melihat, menganalisis, mengedit, dan menyimpan SQL kode yang dikonversi.

Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi SQL kode di C ++, C #, Java, atau kode aplikasi lainnya. Untuk SQL konversi Oracle ke Postgre, Anda dapat menggunakan AWS SCT untuk mengonversi kode SQL *Plus ke. PSQL Juga, untuk SQL konversi Oracle ke Postgre, Anda dapat menggunakan untuk mengonversi SQL kode yang disematkan AWS SCT ke dalam aplikasi C #, C ++, Java, dan Pro*C.

Topik

- [Ikhtisar aplikasi konversi SQL](#)
- [Mengonversi SQL kode dalam aplikasi Anda dengan AWS SCT](#)
- [Mengonversi SQL kode dalam aplikasi C # dengan AWS Schema Conversion Tool](#)
- [Mengonversi SQL kode dalam aplikasi C ++ dengan AWS Schema Conversion Tool](#)
- [Mengonversi SQL kode dalam aplikasi Java dengan AWS Schema Conversion Tool](#)
- [Mengonversi SQL kode dalam aplikasi Pro* C dengan AWS Schema Conversion Tool](#)

Ikhtisar aplikasi konversi SQL

Untuk mengonversi SQL kode dalam aplikasi Anda, lakukan langkah-langkah tingkat tinggi berikut:

- Buat proyek konversi aplikasi - Proyek konversi aplikasi adalah anak dari proyek konversi skema database. Setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi generik di AWS SCT](#).
- Menganalisis dan mengonversi SQL kode Anda — AWS SCT menganalisis aplikasi Anda, mengekstrak SQL kode, dan membuat versi lokal dari konversi SQL untuk Anda tinjau dan edit. Alat ini tidak mengubah kode dalam aplikasi Anda sampai Anda siap. Untuk informasi selengkapnya, lihat [Menganalisis dan mengonversi SQL kode Anda AWS SCT](#).
- Buat laporan penilaian aplikasi — Laporan penilaian aplikasi memberikan informasi penting tentang konversi SQL kode aplikasi dari skema basis data sumber Anda ke skema basis data target Anda.

Untuk informasi selengkapnya, lihat [Membuat dan menggunakan laporan AWS SCT penilaian di AWS SCT](#).

- Edit, terapkan perubahan, dan simpan SQL kode yang dikonversi — Laporan penilaian menyertakan daftar item SQL kode yang tidak dapat dikonversi secara otomatis. Untuk item ini, Anda dapat mengedit SQL kode secara manual untuk melakukan konversi. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan SQL kode konversi Anda dengan AWS SCT](#).

Mengonversi SQL kode dalam aplikasi Anda dengan AWS SCT

Anda dapat menggunakan AWS SCT untuk mengonversi SQL kode yang disematkan ke dalam aplikasi Anda. Konverter AWS SCT aplikasi generik memperlakukan kode aplikasi Anda sebagai teks biasa. Ini memindai kode aplikasi Anda dan mengekstrak SQL kode dengan ekspresi reguler. Konverter ini mendukung berbagai jenis file kode sumber dan bekerja dengan kode aplikasi yang ditulis dalam bahasa pemrograman apa pun.

Konverter aplikasi generik memiliki batasan berikut. Itu tidak menyelam jauh ke dalam logika aplikasi yang khusus untuk bahasa pemrograman aplikasi Anda. Juga, konverter generik tidak mendukung SQL pernyataan dari objek aplikasi yang berbeda, seperti fungsi, parameter, variabel lokal, dan sebagainya.

Untuk meningkatkan konversi SQL kode aplikasi Anda, gunakan konverter SQL kode aplikasi khusus bahasa. Lihat informasi selengkapnya di [SQLkode dalam aplikasi C #](#), [SQLkode di Jawa](#), dan [SQLkode dalam Pro*C](#).

Membuat proyek konversi aplikasi generik di AWS SCT

Dalam AWS Schema Conversion Tool, proyek konversi aplikasi adalah anak dari proyek konversi skema database. Setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak.

Note

AWS SCT tidak mendukung konversi antara sumber dan target berikut:

- Oracle para Oracle
- Postgre SQL ke Postgre atau Aurora Postgre SQL SQL
- Saya SQL ke Saya SQL

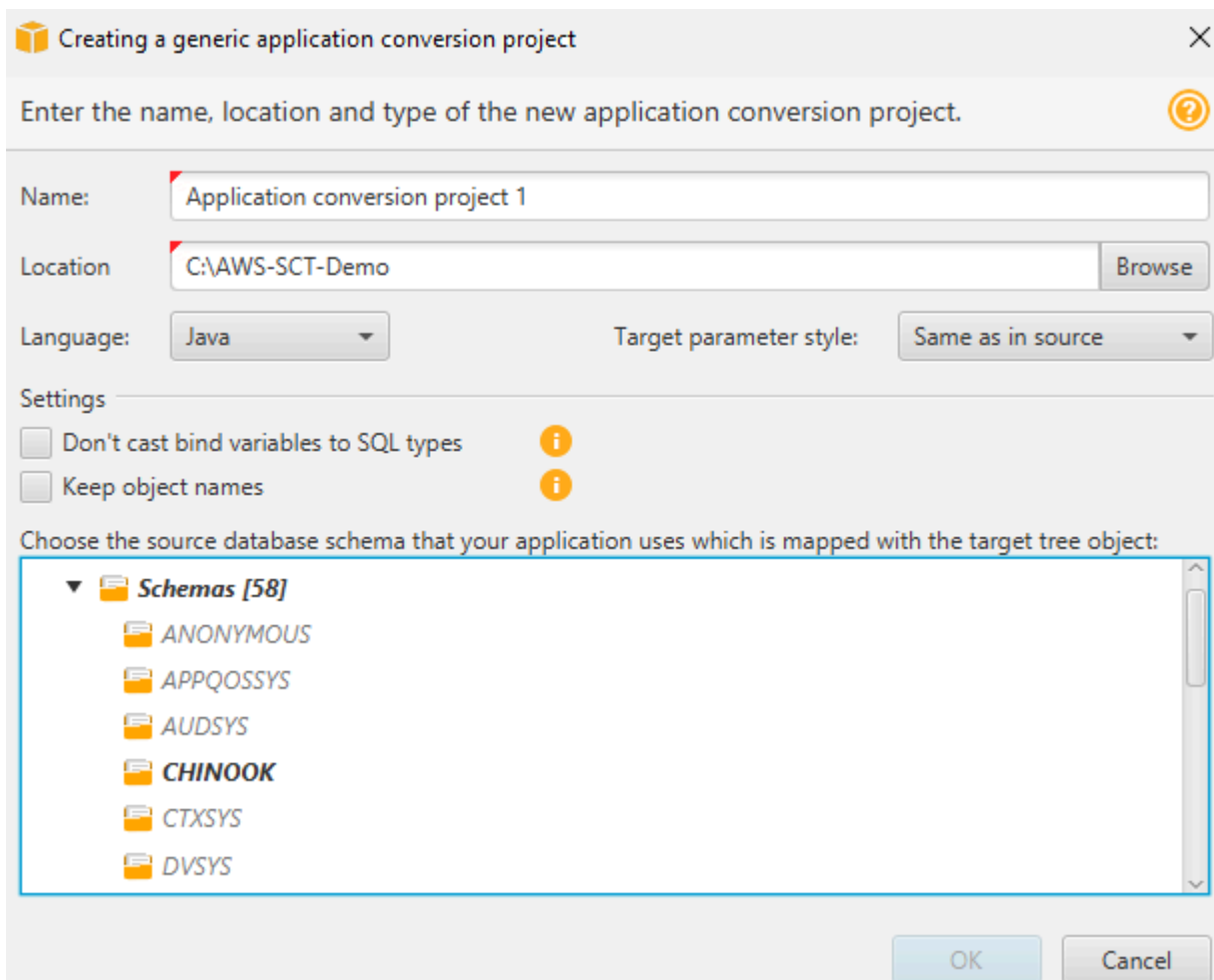
- SQLServer ke SQL Server
- Pergeseran Merah Amazon ke Pergeseran Merah Amazon
- SQLServer ke Babelfish
- SQL Layanan Integrasi Server untuk AWS Glue
- Apache Cassandra ke Amazon DynamoDB

Gunakan prosedur berikut untuk membuat proyek konversi aplikasi generik.

Untuk membuat proyek konversi aplikasi

1. Di dalam AWS Schema Conversion Tool, pilih Aplikasi generik baru pada menu Aplikasi.

Kotak dialog proyek konversi aplikasi baru muncul.



2. Menambahkan informasi proyek berikut.

Untuk parameter ini	Lakukan hal berikut
Nama	Masukkan nama untuk proyek konversi aplikasi Anda. Setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak, jadi pilih nama yang masuk akal jika Anda menambahkan lebih banyak proyek nanti.
Lokasi	Masukkan lokasi kode sumber untuk aplikasi Anda.
Bahasa	Pilih salah satu cara berikut: <ul style="list-style-type: none">• Java• C++• C#• Apa saja
Gaya parameter target	Pilih sintaks yang akan digunakan untuk mengikat variabel dalam kode yang dikonversi. Platform database yang berbeda menggunakan sintaks yang berbeda untuk variabel bind. Pilih salah satu opsi berikut: <ul style="list-style-type: none">• Sama seperti di sumber• Posisi (?)• Diindeks (:1)• Diindeks (\$1)• Dinamakan (@name)• Bernama (:nama)• Bernama (&nama)• Bernama (\$ nama)• Dinamakan (#name)• Dinamakan (! nama!)

Untuk parameter ini	Lakukan hal berikut
Pilih skema basis data sumber	Di pohon sumber, pilih skema yang digunakan aplikasi Anda. Pastikan bahwa skema ini adalah bagian dari aturan pemetaan.

- Pilih Jangan mentransmisikan variabel pengikat ke SQL tipe untuk menghindari konversi jenis variabel pengikat ke SQL tipe. Opsi ini hanya tersedia untuk konversi Oracle ke PostgreSQL.

Misalnya, kode aplikasi sumber Anda menyertakan kueri Oracle berikut:

```
SELECT * FROM ACCOUNT WHERE id = ?
```

Saat Anda memilih Jangan mentransmisikan variabel pengikat ke SQL tipe, AWS SCT konversi kueri ini seperti yang ditunjukkan berikut.

```
SELECT * FROM account WHERE id = ?
```

Saat Anda menghapus Don't cast bind variable to SQL types, AWS SCT ubah tipe variabel bind ke tipe NUMERIC data. Hasil konversi ditunjukkan sebagai berikut.

```
SELECT * FROM account WHERE id = (?)::NUMERIC
```

- Pilih Simpan nama objek untuk menghindari penambahan nama skema ke nama objek yang dikonversi. Opsi ini hanya tersedia untuk konversi Oracle ke PostgreSQL.

Misalnya, misalkan kode aplikasi sumber Anda menyertakan kueri Oracle berikut.

```
SELECT * FROM ACCOUNT
```

Saat Anda memilih Simpan nama objek, AWS SCT konversi kueri ini seperti yang ditunjukkan berikut.

```
SELECT * FROM account
```

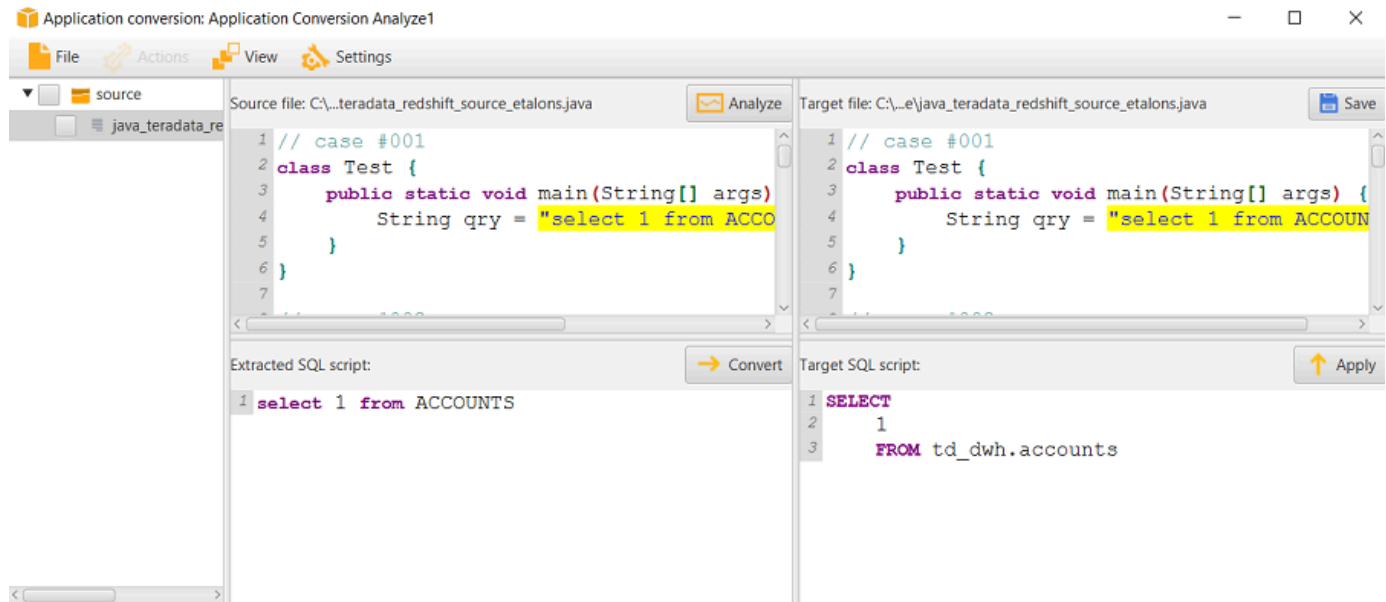
Saat Anda menghapus nama objek Keep, AWS SCT tambahkan nama skema ke nama tabel. Hasil konversi ditunjukkan sebagai berikut.

```
SELECT * FROM schema_name.account
```

Jika kode sumber Anda menyertakan nama objek induk dalam nama objek, AWS SCT gunakan format ini dalam kode yang dikonversi. Dalam hal ini, abaikan opsi Keep object names karena AWS SCT menambahkan nama objek induk dalam kode yang dikonversi.

5. Pilih OK untuk membuat proyek konversi aplikasi Anda.

Jendela proyek terbuka.



Mengelola proyek konversi aplikasi di AWS SCT

Anda dapat membuka proyek konversi aplikasi yang ada dan menambahkan beberapa proyek konversi aplikasi.

Setelah Anda membuat proyek konversi aplikasi, jendela proyek terbuka secara otomatis. Anda dapat menutup jendela proyek konversi aplikasi dan kembali ke sana nanti.

Untuk membuka proyek konversi aplikasi yang ada

1. Di panel kiri, pilih node proyek konversi aplikasi, dan buka menu konteks (klik kanan).
2. Pilih Kelola aplikasi.

Untuk menambahkan proyek konversi aplikasi tambahan

1. Di panel kiri, pilih node proyek konversi aplikasi, dan buka menu konteks (klik kanan).
2. Pilih Aplikasi baru.
3. Masukkan informasi yang diperlukan untuk membuat proyek konversi aplikasi baru. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi generik](#).

Menganalisis dan mengonversi SQL kode Anda AWS SCT

Gunakan prosedur berikut untuk menganalisis dan mengonversi SQL kode Anda di AWS Schema Conversion Tool.

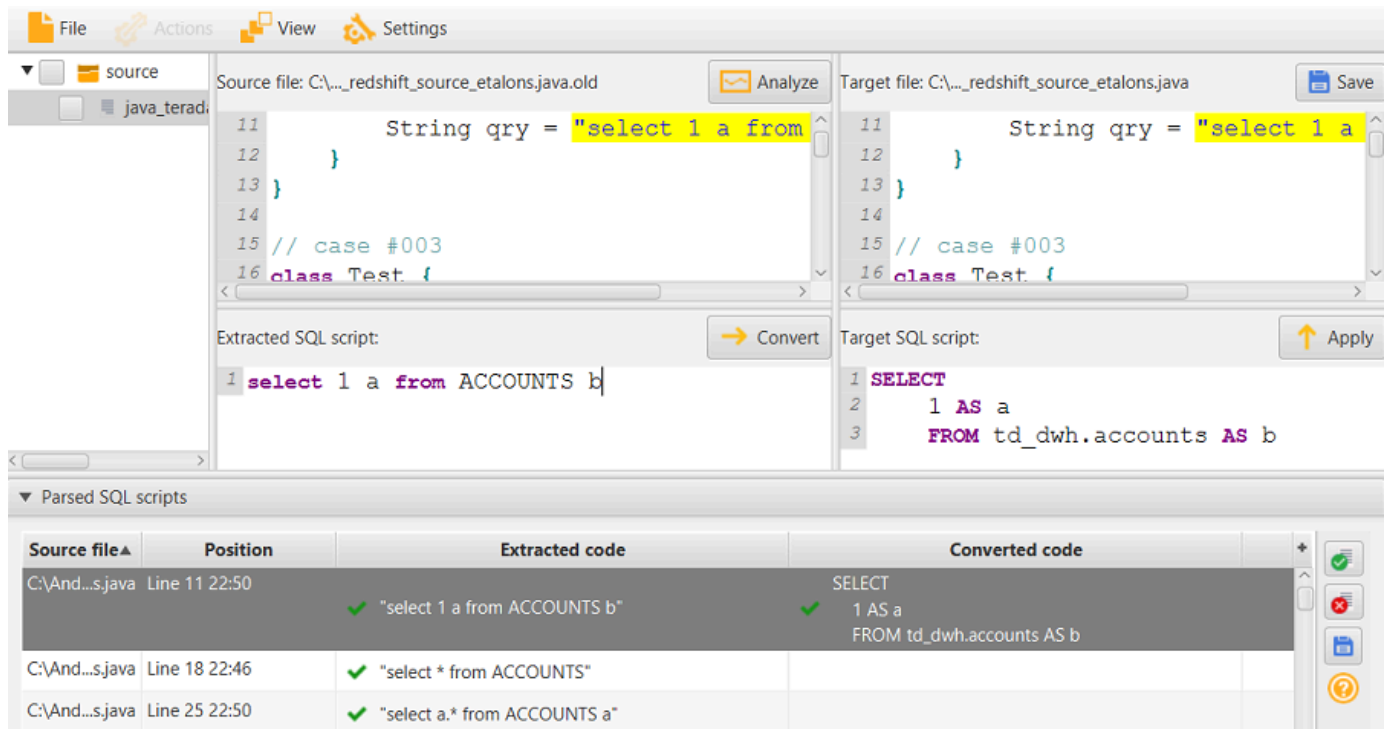
Untuk menganalisis dan mengonversi SQL kode Anda

1. Buka proyek konversi aplikasi yang ada, dan pilih Analisis.

AWS SCT menganalisis kode aplikasi Anda dan mengekstrak SQL kode. AWS SCT menampilkan SQL kode yang diekstrak dalam daftar SQLskrip Parsed.

2. Untuk SQLskrip yang diurai, pilih item untuk meninjau kode yang diekstraksiSQL. AWS SCT menampilkan kode item yang dipilih di panel SQLskrip yang diekstraksi.
3. Pilih Konversi untuk mengonversi SQL kode panel SQLskrip yang diekstrak. AWS SCT mengkonversi kode ke format yang kompatibel dengan database target Anda.

Anda dapat mengedit SQL kode yang dikonversi. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan SQL kode konversi Anda](#).



4. Saat Anda membuat laporan penilaian konversi aplikasi, AWS SCT mengonversi semua item SQL kode yang diekstrak. Untuk informasi selengkapnya, lihat [Membuat dan menggunakan laporan penilaian](#).

Membuat dan menggunakan laporan AWS SCT penilaian di AWS SCT

Laporan penilaian konversi aplikasi memberikan informasi tentang mengonversi SQL kode aplikasi ke format yang kompatibel dengan basis data target Anda. Laporan merinci semua SQL kode yang diekstraksi, semua SQL kode yang dikonversi, dan item tindakan untuk SQL kode yang tidak AWS SCT dapat dikonversi.

Membuat laporan penilaian konversi aplikasi

Gunakan prosedur berikut untuk membuat laporan penilaian konversi aplikasi.

Untuk membuat laporan penilaian konversi aplikasi

1. Di jendela proyek konversi aplikasi, pilih Buat laporan pada menu Tindakan.

AWS SCT membuat laporan penilaian konversi aplikasi dan membukanya di jendela proyek konversi aplikasi.

2. Tinjau tab Ringkasan.

Tab Ringkasan, ditampilkan berikut, menampilkan informasi ringkasan dari laporan penilaian aplikasi. Ini menunjukkan item SQL kode yang dikonversi secara otomatis, dan item yang tidak dikonversi secara otomatis.



3. Pilih tindakan SQL ekstraksi.

Tinjau daftar item SQL kode yang tidak AWS SCT dapat mengekstrak dari kode sumber Anda.

4. Pilih tindakan SQL konversi.

Tinjau daftar item SQL kode yang tidak AWS SCT dapat dikonversi secara otomatis. Gunakan tindakan yang disarankan untuk mengonversi SQL kode secara manual. Untuk informasi tentang cara mengedit SQL kode yang dikonversi, lihat [Mengedit dan menyimpan SQL kode konversi Anda dengan AWS SCT](#).

5. (Opsional) Simpan salinan lokal laporan sebagai PDF file atau file values (CSV) yang dipisahkan koma:

- Pilih Simpan ke PDF kanan atas untuk menyimpan laporan sebagai PDF file.

PDFFile berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi aplikasi.

- Pilih Simpan ke CSV kanan atas untuk menyimpan laporan sebagai CSV file.

CSVFile berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi SQL kode.

Mengedit dan menyimpan SQL kode konversi Anda dengan AWS SCT

Laporan penilaian menyertakan daftar item SQL kode yang tidak AWS SCT dapat dikonversi. Untuk setiap item, AWS SCT buat item tindakan di tab tindakan SQL konversi. Untuk item ini, Anda dapat mengedit SQL kode secara manual untuk melakukan konversi.

Gunakan prosedur berikut untuk mengedit SQL kode konversi Anda, menerapkan perubahan, dan kemudian menyimpannya.

Untuk mengedit, menerapkan perubahan ke, dan menyimpan SQL kode yang telah dikonversi

1. Edit SQL kode Anda yang dikonversi secara langsung di panel SQLskrip Target. Jika tidak ada kode yang dikonversi yang ditampilkan, Anda dapat mengklik panel dan mulai mengetik.
2. Setelah Anda selesai mengedit SQL kode yang dikonversi, pilih Terapkan. Pada titik ini, perubahan disimpan dalam memori, tetapi belum ditulis ke file Anda.
3. Pilih Simpan untuk menyimpan perubahan ke file Anda.

Ketika Anda memilih Simpan, Anda menimpa file asli Anda. Buat salinan file asli Anda sebelum menyimpan sehingga Anda memiliki catatan kode aplikasi asli Anda.

Mengonversi SQL kode dalam aplikasi C # dengan AWS Schema Conversion Tool

Untuk SQL konversi Oracle ke Postgre, Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi SQL kode yang disematkan ke dalam aplikasi C # Anda. Konverter aplikasi C # khusus ini memahami logika aplikasi. Ini mengumpulkan pernyataan yang terletak di objek aplikasi yang berbeda, seperti fungsi, parameter, variabel lokal, dan sebagainya.

Karena analisis mendalam ini, konverter SQL kode aplikasi C # memberikan hasil konversi yang lebih baik daripada konverter generik.

Membuat proyek konversi aplikasi C # di AWS SCT

Anda dapat membuat proyek konversi aplikasi C # hanya untuk mengonversi skema database Oracle ke skema database Postgre. Pastikan Anda menambahkan aturan pemetaan dalam proyek Anda yang menyertakan skema Oracle sumber dan database Postgre target. Untuk informasi selengkapnya, lihat [Memetakan tipe data di AWS Schema Conversion Tool](#).

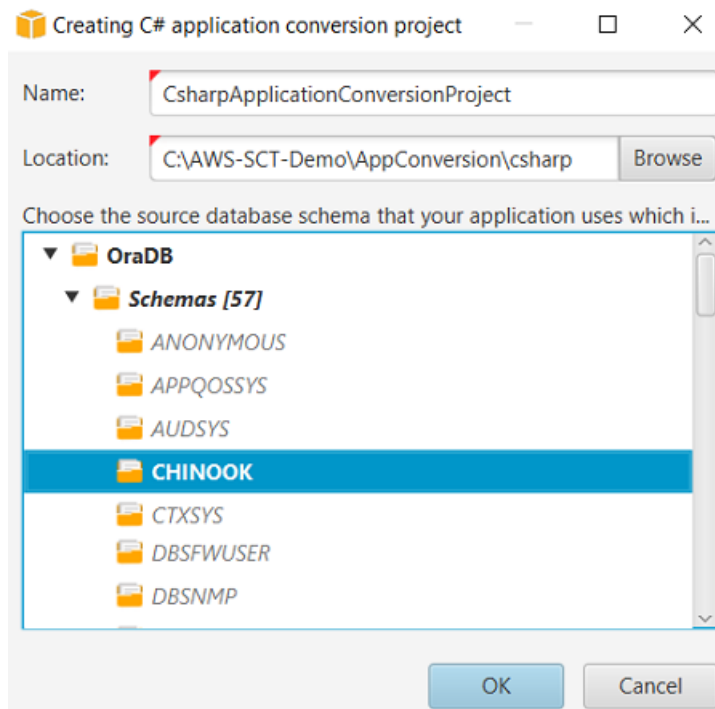
Anda dapat menambahkan beberapa proyek konversi aplikasi dalam satu AWS SCT proyek. Gunakan prosedur berikut untuk membuat proyek konversi aplikasi C #.

Untuk membuat proyek konversi aplikasi C #

1. Buat proyek konversi database, dan tambahkan database Oracle sumber. Untuk informasi selengkapnya, silakan lihat [Memulai dan mengelola Proyek di AWS SCT](#) dan [Menambahkan server ke proyek di AWS SCT](#).

2. Tambahkan aturan pemetaan yang mencakup database Oracle sumber Anda dan database SQL Postgre target. Anda dapat menambahkan database Postgre target atau menggunakan platform SQL database SQL target Postgre virtual dalam aturan pemetaan. Untuk informasi selengkapnya, silakan lihat [Memetakan tipe data di AWS Schema Conversion Tool](#) dan [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).
3. Pada menu Tampilan, pilih Tampilan utama.
4. Pada menu Aplikasi, pilih aplikasi C# Baru.

Kotak dialog Creating a C# application conversion project muncul.



5. Untuk Nama, masukkan nama untuk proyek konversi aplikasi C # Anda. Karena setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak, pilih nama yang masuk akal jika Anda menambahkan beberapa proyek.
6. Untuk Lokasi, masukkan lokasi kode sumber untuk aplikasi Anda.
7. Di pohon sumber, pilih skema yang digunakan aplikasi Anda. Pastikan bahwa skema ini adalah bagian dari aturan pemetaan. AWS SCT menyoroti skema yang merupakan bagian dari aturan pemetaan dalam huruf tebal.
8. Pilih OK untuk membuat proyek konversi aplikasi C # Anda.
9. Temukan proyek konversi aplikasi C # Anda di node Applications di panel kiri.

Mengonversi kode aplikasi SQL C # Anda AWS SCT

Setelah Anda menambahkan aplikasi C# Anda ke AWS SCT proyek, konversi SQL kode dari aplikasi ini ke format yang kompatibel dengan platform database target Anda. Gunakan prosedur berikut untuk menganalisis dan mengonversi SQL kode yang disematkan dalam aplikasi C# Anda di. AWS Schema Conversion Tool

Untuk mengonversi SQL kode Anda

1. Perluas node C # di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan dikonversi, dan buka menu konteks (klik kanan).
3. Pilih Konversi. AWS SCT menganalisis file kode sumber Anda, menentukan logika aplikasi, dan memuat metadata kode ke dalam proyek. Metadata kode ini mencakup kelas C #, objek, metode, variabel global, antarmuka, dan sebagainya.

Di panel database target, AWS SCT buat struktur folder yang mirip dengan proyek aplikasi sumber Anda. Di sini Anda dapat meninjau kode aplikasi yang dikonversi.

```
Source Oracle file: SpecialEscapeSequences.cs
Properties Text Related converted objects Statistics Settings
18 {
19     string str1 = "SELECT *\n" +
20                 "FROM\t JAVA DB.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 JAVA DB.DATETYPE_MIXED_ALL\n\r" +
22                 "WHERE COL_CHAR = \ 'CHAR\ ' ";
23
24     command.CommandText = str1;
25     command.ExecuteNonQuery();
26 }
27 connection.Close();
```

4. Simpan kode aplikasi Anda yang telah dikonversi. Untuk informasi selengkapnya, lihat [Menyimpan kode aplikasi Anda yang telah dikonversi](#).

Aplikasi C# Anda mungkin menyertakan SQL kode yang berinteraksi dengan basis data sumber yang berbeda. Anda dapat bermigrasi ke Postgre SQL beberapa database sumber ini. Dalam hal ini, pastikan Anda tidak mengonversi SQL kode yang berinteraksi dengan database yang dikecualikan dari cakupan migrasi. Anda dapat mengecualikan file sumber aplikasi C# Anda dari lingkup konversi. Untuk melakukannya, kosongkan kotak centang untuk nama file yang ingin Anda kecualikan dari lingkup konversi.

Setelah Anda mengubah cakupan konversi, AWS SCT masih menganalisis SQL kode semua file sumber aplikasi C# Anda. Kemudian, AWS SCT salin ke folder target semua file sumber yang Anda kecualikan dari lingkup konversi. Operasi ini memungkinkan untuk membangun aplikasi Anda setelah Anda menyimpan file aplikasi yang dikonversi.

Menyimpan kode aplikasi Anda yang dikonversi dengan AWS SCT

Gunakan prosedur berikut untuk menyimpan kode aplikasi Anda yang telah dikonversi.

Untuk menyimpan kode aplikasi yang telah dikonversi

1. Perluas node C # di bawah Aplikasi di panel database target.
2. Pilih aplikasi yang dikonversi, dan pilih Simpan.
3. Masukkan path ke folder untuk menyimpan kode aplikasi yang dikonversi, dan pilih Pilih folder.

Mengelola proyek konversi aplikasi C # di AWS SCT

Anda dapat menambahkan beberapa proyek konversi aplikasi C#, memperbarui kode aplikasi dalam AWS SCT proyek, atau menghapus proyek konversi C# dari proyek Anda AWS SCT .

Untuk menambahkan proyek konversi aplikasi C# tambahan

1. Perluas node Applications di panel kiri.
2. Pilih node C #, dan buka menu konteks (klik kanan).
3. Pilih Aplikasi baru.
4. Masukkan informasi yang diperlukan untuk membuat proyek konversi aplikasi C # baru. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi C #](#).

Setelah Anda membuat perubahan dalam kode aplikasi sumber Anda, unggah ke dalam AWS SCT proyek.

Untuk mengunggah kode aplikasi yang diperbarui

1. Perluas node C # di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan diperbarui, dan buka menu konteks (klik kanan).
3. Pilih Refresh dan kemudian pilih Yes.

AWS SCT mengunggah kode aplikasi Anda dari file sumber dan menghapus hasil konversi. Untuk menyimpan perubahan kode yang Anda buat AWS SCT dan hasil konversi, buat proyek konversi C# baru.

Untuk menghapus proyek konversi aplikasi C #

1. Perluas node C # di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus dan kemudian pilih OK.

Membuat laporan penilaian konversi aplikasi C# di AWS SCT

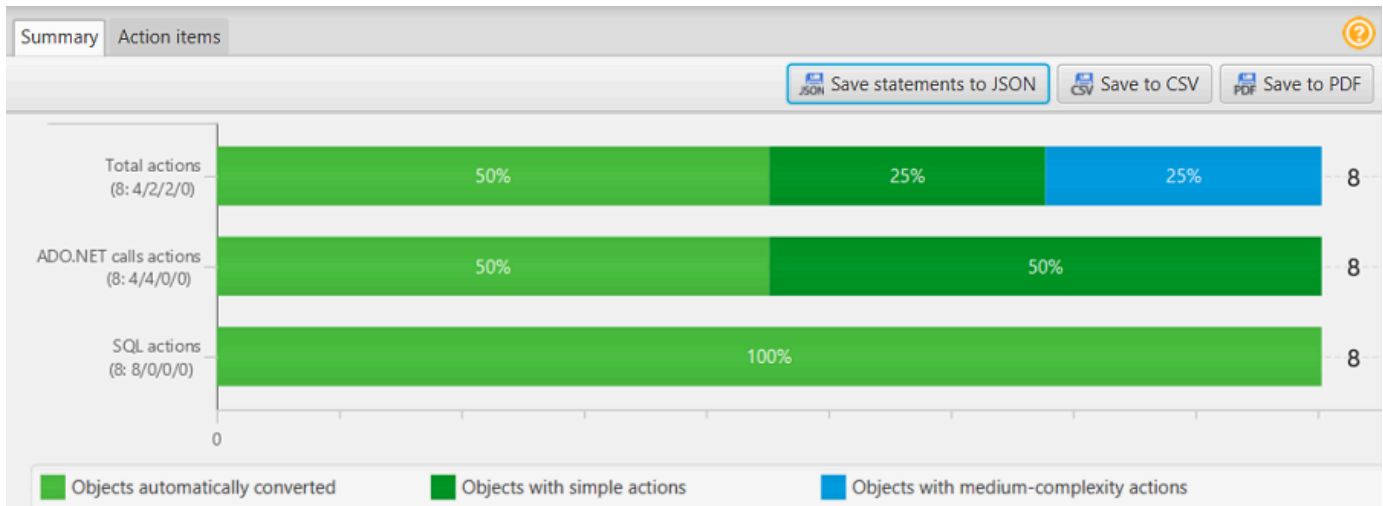
Laporan penilaian konversi aplikasi C# memberikan informasi tentang mengonversi SQL kode yang disematkan dalam aplikasi C# Anda ke format yang kompatibel dengan basis data target Anda. Laporan penilaian memberikan detail konversi untuk semua titik SQL eksekusi dan semua file kode sumber. Laporan penilaian juga menyertakan item tindakan untuk SQL kode yang tidak AWS SCT dapat dikonversi.

Gunakan prosedur berikut untuk membuat laporan penilaian konversi aplikasi C#.

Untuk membuat laporan penilaian konversi aplikasi C#

1. Perluas node C # di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan dikonversi, dan buka menu konteks (klik kanan).
3. Pilih Konversi.
4. Pada menu Lihat, pilih Tampilan laporan penilaian.
5. Lihat tab Ringkasan.

Tab Ringkasan, ditampilkan berikut, menampilkan informasi ringkasan eksekutif dari laporan penilaian aplikasi C#. Ini menunjukkan hasil konversi untuk semua titik SQL eksekusi dan semua file kode sumber.



6. Pilih Simpan pernyataan JSON untuk menyimpan SQL kode yang diekstrak dari aplikasi C# Anda sebagai JSON file.
7. (Opsional) Simpan salinan lokal laporan sebagai PDF file atau file values (CSV) yang dipisahkan koma:
 - Pilih Simpan ke PDF kanan atas untuk menyimpan laporan sebagai PDF file.
PDFFile berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi aplikasi.
 - Pilih Simpan ke CSV kanan atas untuk menyimpan laporan sebagai CSV file.
CSVFile berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi SQL kode.

Mengonversi SQL kode dalam aplikasi C ++ dengan AWS Schema Conversion Tool

Untuk SQL konversi Oracle ke Postgre, Anda dapat menggunakan AWS SCT untuk mengonversi SQL kode yang disematkan ke dalam aplikasi C ++ Anda. Konverter aplikasi C ++ khusus ini memahami logika aplikasi. Ini mengumpulkan pernyataan yang terletak di objek aplikasi yang berbeda, seperti fungsi, parameter, variabel lokal, dan sebagainya.

Karena analisis mendalam ini, konverter SQL kode aplikasi C ++ memberikan hasil konversi yang lebih baik daripada konverter generik.

Membuat proyek konversi aplikasi C ++ di AWS SCT

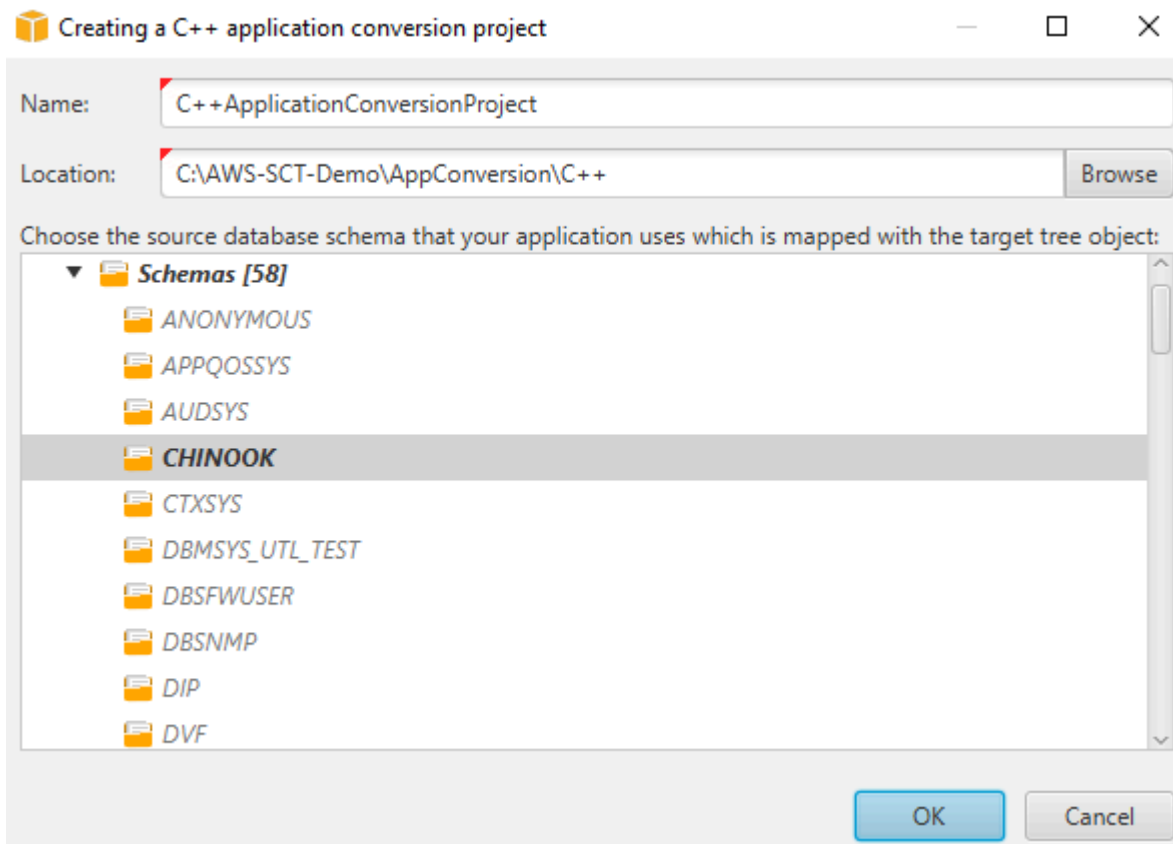
Anda dapat membuat proyek konversi aplikasi C ++ hanya untuk mengonversi skema database Oracle ke skema database Postgre. SQL Pastikan Anda menambahkan aturan pemetaan dalam proyek Anda yang menyertakan skema Oracle sumber dan database Postgre target. SQL Untuk informasi selengkapnya, lihat [Memetakan tipe data di AWS Schema Conversion Tool](#).

Anda dapat menambahkan beberapa proyek konversi aplikasi dalam satu AWS SCT proyek.

Untuk membuat proyek konversi aplikasi C ++

1. Buat proyek konversi database, dan tambahkan database Oracle sumber. Untuk informasi selengkapnya, silakan lihat [Memulai dan mengelola Proyek di AWS SCT](#) dan [Menambahkan server ke proyek di AWS SCT](#).
2. Tambahkan aturan pemetaan yang mencakup database Oracle sumber Anda dan database SQL Postgre target. Anda dapat menambahkan database Postgre target atau menggunakan platform SQL database SQL target Postgre virtual dalam aturan pemetaan. Untuk informasi selengkapnya, silakan lihat [Memetakan tipe data di AWS Schema Conversion Tool](#) dan [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).
3. Pada menu Tampilan, pilih Tampilan utama.
4. Pada menu Aplikasi, pilih aplikasi C++ Baru.

Kotak dialog Creating a C ++ application conversion project muncul.



5. Untuk Nama, masukkan nama untuk proyek konversi aplikasi C ++ Anda. Karena setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak, pilih nama yang masuk akal jika Anda menambahkan beberapa proyek.
6. Untuk Lokasi, masukkan lokasi kode sumber untuk aplikasi Anda.
7. Di pohon sumber, pilih skema yang digunakan aplikasi Anda. Pastikan bahwa skema ini adalah bagian dari aturan pemetaan. AWS SCT menyoroti skema yang merupakan bagian dari aturan pemetaan dalam huruf tebal.
8. Pilih OK untuk membuat proyek konversi aplikasi C ++ Anda.
9. Temukan proyek konversi aplikasi C ++ Anda di node Applications di panel kiri.

Mengonversi kode aplikasi SQL C ++ Anda di AWS SCT

Setelah Anda menambahkan aplikasi C++ Anda ke AWS SCT proyek, konversi SQL kode dari aplikasi ini ke format yang kompatibel dengan platform database target Anda. Gunakan prosedur berikut untuk menganalisis dan mengonversi SQL kode yang disematkan dalam AWS SCT aplikasi C ++ Anda.


Untuk mengonversi SQL kode Anda

1. Perluas node C++ di bawah Aplikasi di panel kiri, dan pilih aplikasi yang akan dikonversi.
2. Dalam proyek aplikasi Source Oracle, pilih Pengaturan. Tinjau dan edit pengaturan konversi untuk aplikasi C++ yang dipilih. Anda juga dapat menentukan pengaturan konversi untuk semua aplikasi C++ yang Anda tambahkan ke AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Mengelola proyek konversi aplikasi C++](#).
3. Untuk jenis Compiler, pilih compiler yang Anda gunakan untuk kode sumber aplikasi C++ Anda. AWS SCT mendukung compiler C++ berikut: Microsoft Visual C++, GNUCompiler CollectionGCC, dan Clang. Opsi defaultnya adalah Microsoft Visual C++.
4. Untuk makro yang ditentukan pengguna, masukkan jalur ke file yang menyertakan makro yang ditentukan pengguna dari proyek C++ Anda. Pastikan bahwa file ini memiliki struktur berikut:
`#define name value` Dalam contoh sebelumnya, `value` adalah parameter opsional. Nilai default untuk parameter opsional ini adalah `1`.

Untuk membuat file ini, buka proyek Anda di Microsoft Visual Studio, lalu pilih Project, Properties, C/C++, dan Preprocessor. Untuk definisi Preprocessor, pilih Edit dan salin nama dan nilai ke file teks baru. Kemudian, untuk setiap string dalam file, tambahkan awalan berikut: `#define` .

5. Untuk direktori External include, masukkan jalur ke folder yang menyertakan pustaka eksternal yang Anda gunakan dalam proyek C++ Anda.
6. Di panel kiri, pilih aplikasi yang akan dikonversi, dan buka menu konteks (klik kanan).
7. Pilih Konversi. AWS SCT menganalisis file kode sumber Anda, menentukan logika aplikasi, dan memuat metadata kode ke dalam proyek. Metadata kode ini mencakup kelas C++, objek, metode, variabel global, antarmuka, dan sebagainya.

Di panel database target, AWS SCT buat struktur folder yang mirip dengan proyek aplikasi sumber Anda. Di sini Anda dapat meninjau kode aplikasi yang dikonversi, seperti yang ditunjukkan berikut.



The screenshot displays two panels of the AWS SCT interface. The top panel, titled 'Source Oracle file: StringInitialization.cpp', shows the original C++ code. The bottom panel, titled 'Target Amazon RDS for PostgreSQL file: StringInitialization.cpp', shows the converted code. The converted code replaces Oracle-specific functions like `SQLDriverConnect`, `SQLAllocHandle`, and `SQLExecDirect` with their PostgreSQL equivalents: `PGconn`, `PQconnectdb`, and `PQexec`. The SQL query is also converted from Oracle's `SELECT JAVADB.GET_INT() FROM DUAL` to PostgreSQL's `SELECT javadb.get_int()`.

```
44     if ((dRet == SQLDriverConnect(hDBC, NULL, lpConnectionStr, connectionStr.size(), OutConnStr, 0x00000000, 0x00000000, 0x00000000, 0x00000000)) == SQL_SUCCESS)
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             {
```

```
45     if ((dRet == PGconn * pConn = PQconnectdb(lpConnectionStr)) == 0)
46     {
47         SQLHANDLE hSelectStm = NULL;
48
49         if ((dRet = PQconnectdb(lpConnectionStr)) == 0)
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 = PQexec(pConn, buff)) == 0)
56             {
```

8. Simpan kode aplikasi Anda yang telah dikonversi. Untuk informasi selengkapnya, lihat [Menyimpan kode aplikasi Anda yang telah dikonversi](#).

Menyimpan kode aplikasi Anda yang dikonversi dengan AWS SCT

Gunakan prosedur berikut untuk menyimpan kode aplikasi Anda yang telah dikonversi.

Untuk menyimpan kode aplikasi yang telah dikonversi

1. Perluas node C++ di bawah Applications di panel database target.
2. Pilih aplikasi yang dikonversi, dan pilih Simpan.
3. Masukkan path ke folder untuk menyimpan kode aplikasi yang dikonversi, dan pilih Pilih folder.

Mengelola proyek konversi aplikasi C++ di AWS SCT

Anda dapat menambahkan beberapa proyek konversi aplikasi C++, mengedit pengaturan konversi, memperbarui kode aplikasi C++, atau menghapus proyek konversi C++ dari proyek Anda. AWS SCT

Untuk menambahkan proyek konversi aplikasi C++ tambahan

1. Perluas node Applications di panel kiri.
2. Pilih node C ++, dan buka menu konteks (klik kanan).
3. Pilih Aplikasi baru.
4. Masukkan informasi yang diperlukan untuk membuat proyek konversi aplikasi C ++ baru. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi C ++](#).

Anda dapat menentukan pengaturan konversi untuk semua proyek konversi aplikasi C ++ dalam AWS SCT proyek Anda.

Untuk mengedit pengaturan konversi untuk semua aplikasi C++

1. Pada menu Pengaturan, pilih Pengaturan proyek, lalu pilih Konversi aplikasi.
2. Untuk jenis Compiler, pilih compiler yang Anda gunakan untuk kode sumber aplikasi C++ Anda. AWS SCT mendukung compiler C ++ berikut: Microsoft Visual C ++,, GNUCompiler CollectionGCC, dan Clang. Opsi defaultnya adalah Microsoft Visual C ++.
3. Untuk makro yang ditentukan pengguna, masukkan jalur ke file yang menyertakan makro yang ditentukan pengguna dari proyek C++ Anda. Pastikan bahwa file ini memiliki struktur berikut:
`#define name value` Dalam contoh sebelumnya, `value` adalah parameter opsional. Nilai default untuk parameter opsional ini adalah `1`.

Untuk membuat file ini, buka proyek Anda di Microsoft Visual Studio, lalu pilih Project, Properties, C/C++, dan Preprocessor. Untuk definisi Preprocessor, pilih Edit dan salin nama dan nilai ke file teks baru. Kemudian, untuk setiap string dalam file, tambahkan awalan berikut: `#define` .

4. Untuk direktori External include, masukkan jalur ke folder yang menyertakan pustaka eksternal yang Anda gunakan dalam proyek C++ Anda.
5. Pilih OK untuk menyimpan pengaturan proyek dan menutup jendela.

Atau Anda dapat menentukan pengaturan konversi untuk setiap proyek konversi aplikasi C ++. Untuk informasi selengkapnya, lihat [Mengonversi kode aplikasi C++ Anda SQL](#).

Setelah Anda membuat perubahan dalam kode aplikasi sumber Anda, unggah ke dalam AWS SCT proyek.

Untuk mengunggah kode aplikasi yang diperbarui

1. Perluas node C ++ di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan diperbarui, dan buka menu konteks (klik kanan).
3. Pilih Refresh dan kemudian pilih Yes.

AWS SCT mengunggah kode aplikasi Anda dari file sumber dan menghapus hasil konversi. Untuk menyimpan perubahan kode yang Anda buat AWS SCT dan hasil konversi, buat proyek konversi C++ baru.

Juga, AWS SCT menghapus pengaturan konversi aplikasi yang Anda tentukan untuk aplikasi yang dipilih. Setelah Anda mengunggah kode aplikasi yang diperbarui, AWS SCT menerapkan nilai default dari pengaturan proyek.

Untuk menghapus proyek konversi aplikasi C ++

1. Perluas node C ++ di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus dan kemudian pilih OK.

Membuat laporan penilaian konversi aplikasi C++ di AWS SCT

Laporan penilaian konversi aplikasi C ++ memberikan informasi tentang mengonversi SQL kode yang disematkan dalam aplikasi C++ Anda ke format yang kompatibel dengan basis data target Anda. Laporan penilaian memberikan detail konversi untuk semua titik SQL eksekusi dan semua file kode sumber. Laporan penilaian juga menyertakan item tindakan untuk SQL kode yang tidak AWS SCT dapat dikonversi.

Untuk membuat laporan penilaian konversi aplikasi C++

1. Perluas node C ++ di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan dikonversi, dan buka menu konteks (klik kanan).
3. Pilih Konversi.
4. Pada menu Lihat, pilih Tampilan laporan penilaian.
5. Lihat tab Ringkasan.

Tab Ringkasan menampilkan informasi ringkasan eksekutif dari laporan penilaian aplikasi C++. Ini menunjukkan hasil konversi untuk semua titik SQL eksekusi dan semua file kode sumber.

6. Pilih Simpan pernyataan JSON untuk menyimpan SQL kode yang diekstrak dari aplikasi Java Anda sebagai JSON file.
7. (Opsional) Simpan salinan lokal laporan sebagai PDF file atau file values (CSV) yang dipisahkan koma:

- Pilih Simpan ke PDF kanan atas untuk menyimpan laporan sebagai PDF file.

PDFFile berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi aplikasi.

- Pilih Simpan ke CSV kanan atas untuk menyimpan laporan sebagai CSV file.

CSVFile berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi SQL kode.

Mengonversi SQL kode dalam aplikasi Java dengan AWS Schema Conversion Tool

Untuk SQL konversi Oracle ke Postgre, Anda dapat menggunakan AWS Schema Conversion Tool untuk mengonversi SQL kode yang disematkan ke dalam aplikasi Java Anda. Konverter aplikasi Java khusus ini memahami logika aplikasi. Ini mengumpulkan pernyataan yang terletak di objek aplikasi yang berbeda, seperti fungsi, parameter, variabel lokal, dan sebagainya.

Karena analisis mendalam ini, konverter SQL kode aplikasi Java memberikan hasil konversi yang lebih baik dibandingkan dengan konverter generik.

Jika aplikasi Java Anda menggunakan MyBatis kerangka kerja untuk berinteraksi dengan database, maka Anda dapat menggunakannya AWS SCT untuk mengonversi SQL pernyataan yang disematkan ke dalam MyBatis XML file dan anotasi. Untuk memahami logika SQL pernyataan ini, AWS SCT gunakan file MyBatis konfigurasi. AWS SCT dapat secara otomatis menemukan file ini di folder aplikasi Anda, atau Anda dapat memasukkan jalur ke file ini secara manual.

Membuat proyek konversi aplikasi Java di AWS SCT

Anda dapat membuat proyek konversi aplikasi Java hanya untuk mengonversi skema database Oracle ke skema database SQL Postgre. Pastikan Anda menambahkan aturan pemetaan dalam

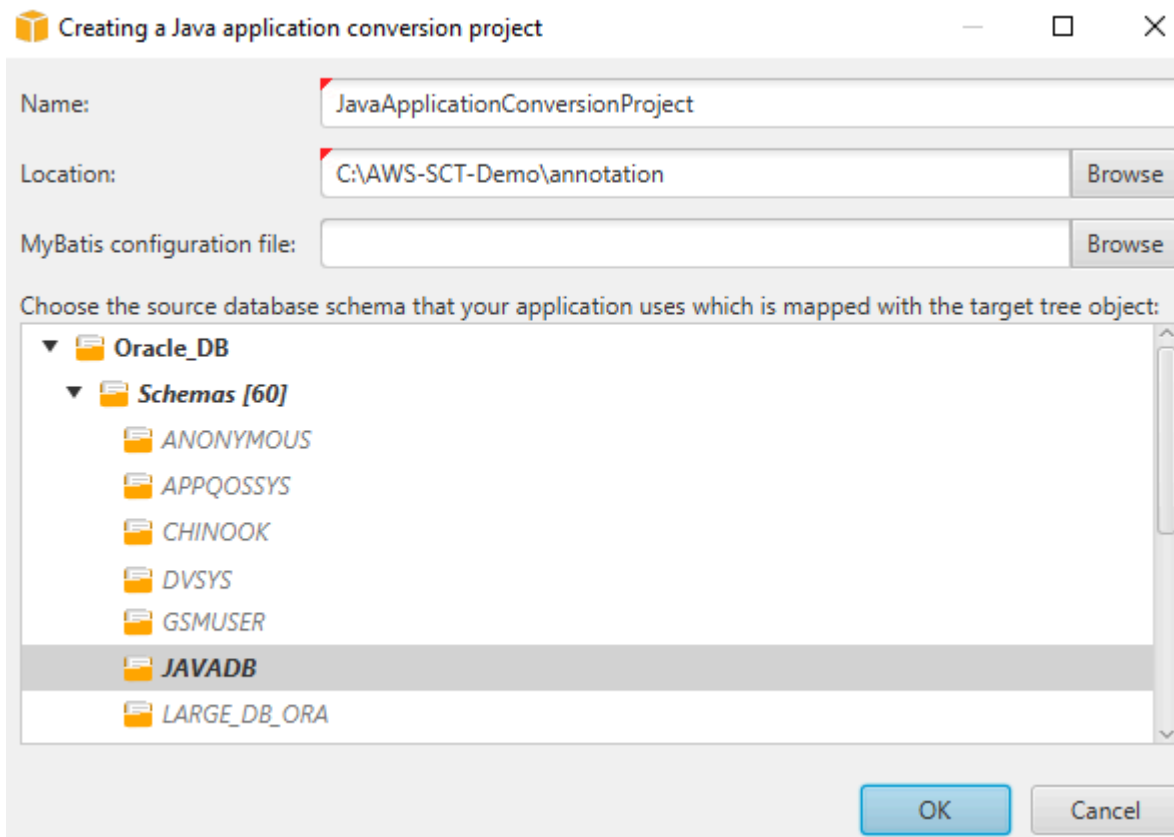
proyek Anda yang menyertakan skema Oracle sumber dan database Postgre target. SQL Untuk informasi selengkapnya, lihat [Memetakan tipe data di AWS Schema Conversion Tool](#).

Anda dapat menambahkan beberapa proyek konversi aplikasi dalam satu AWS SCT proyek. Gunakan prosedur berikut untuk membuat proyek konversi aplikasi Java.

Untuk membuat proyek konversi aplikasi Java

1. Buat proyek konversi database, dan tambahkan database Oracle sumber. Untuk informasi selengkapnya, silakan lihat [Memulai dan mengelola Proyek di AWS SCT](#) dan [Menambahkan server ke proyek di AWS SCT](#).
2. Tambahkan aturan pemetaan yang mencakup database Oracle sumber Anda dan database SQL Postgre target. Anda dapat menambahkan database Postgre target atau menggunakan platform SQL database SQL target Postgre virtual dalam aturan pemetaan. Untuk informasi selengkapnya, silakan lihat [Memetakan tipe data di AWS Schema Conversion Tool](#) dan [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).
3. Pada menu Tampilan, pilih Tampilan utama.
4. Pada menu Applications, pilih New Java Application.

Kotak dialog Creating a Java application conversion project muncul.



5. Untuk Nama, masukkan nama untuk proyek konversi aplikasi Java Anda. Karena setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak, pilih nama yang masuk akal jika Anda menambahkan beberapa proyek.
6. Untuk Lokasi, masukkan lokasi kode sumber untuk aplikasi Anda.
7. (Opsional) Untuk file MyBatis konfigurasi, masukkan path ke file MyBatis konfigurasi. AWS SCT memindai folder aplikasi Anda untuk menemukan file ini secara otomatis. Jika file ini tidak terletak di folder aplikasi Anda, atau jika Anda menggunakan beberapa file konfigurasi, maka masukkan path secara manual.
8. Di pohon sumber, pilih skema yang digunakan aplikasi Anda. Pastikan bahwa skema ini adalah bagian dari aturan pemetaan. AWS SCT menyoroti skema yang merupakan bagian dari aturan pemetaan dalam huruf tebal.
9. Pilih OK untuk membuat proyek konversi aplikasi Java Anda.
10. Temukan proyek konversi aplikasi Java Anda di node Applications di panel kiri.

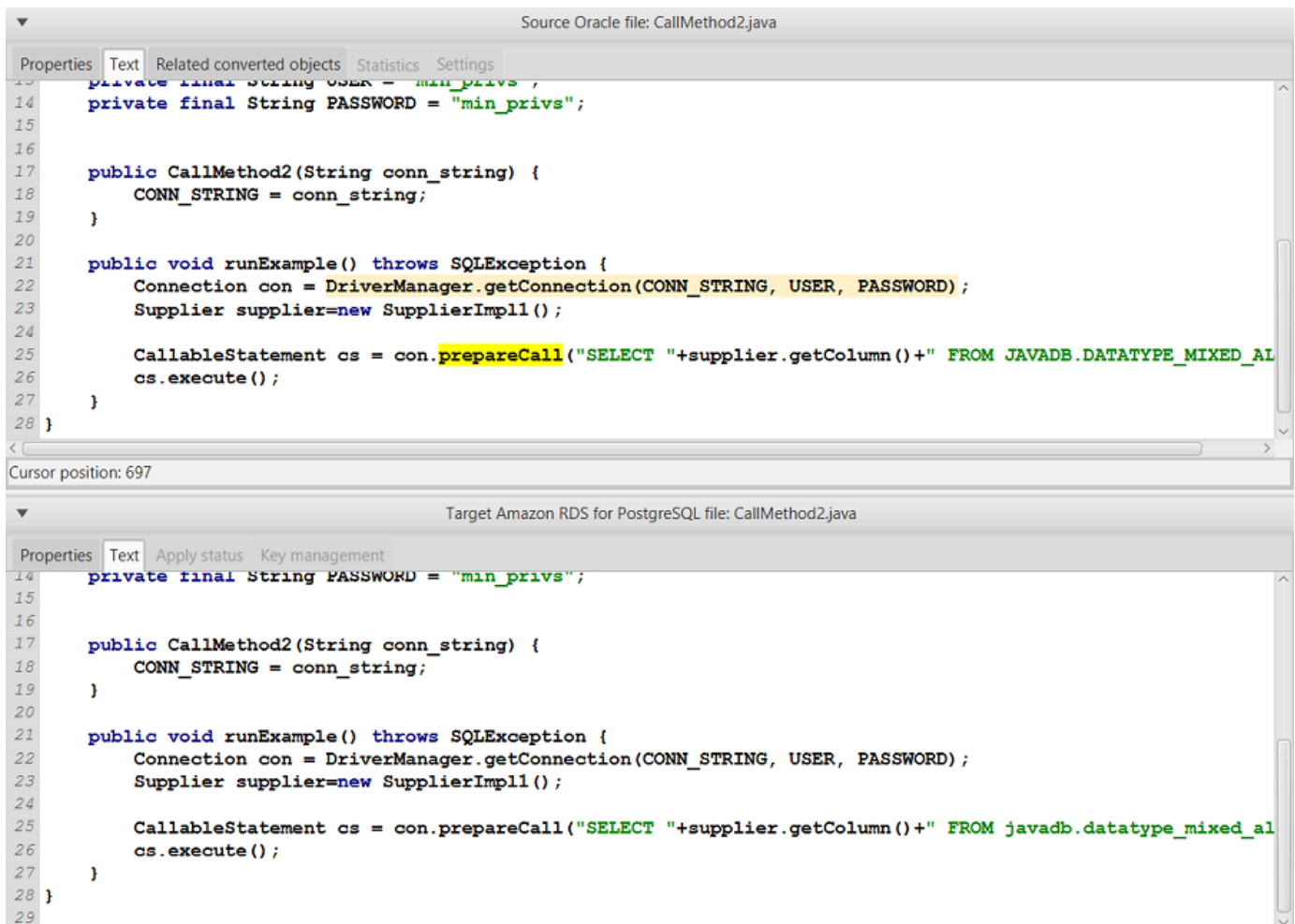
Mengonversi SQL kode aplikasi Java Anda di AWS SCT

Setelah Anda menambahkan aplikasi Java Anda ke AWS SCT proyek, konversi SQL kode dari aplikasi ini ke format yang kompatibel dengan platform database target Anda. Gunakan prosedur berikut untuk menganalisis dan mengonversi SQL kode yang disematkan dalam aplikasi Java Anda di file AWS Schema Conversion Tool.

Untuk mengonversi SQL kode Anda

1. Perluas node Java di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan dikonversi, dan buka menu konteks (klik kanan).
3. Pilih Konversi. AWS SCT menganalisis file kode sumber Anda, menentukan logika aplikasi, dan memuat metadata kode ke dalam proyek. Metadata kode ini mencakup kelas Java, objek, metode, variabel global, antarmuka, dan sebagainya.

Di panel database target, AWS SCT buat struktur folder yang mirip dengan proyek aplikasi sumber Anda. Di sini Anda dapat meninjau kode aplikasi yang dikonversi.



```
Source Oracle file: CallMethod2.java
13 private final String USER = "min_privs";
14 private final String PASSWORD = "min_privs";
15
16
17 public CallMethod2(String conn_string) {
18     CONN_STRING = conn_string;
19 }
20
21 public void runExample() throws SQLException {
22     Connection con = DriverManager.getConnection(CONN_STRING, USER, PASSWORD);
23     Supplier supplier=new SupplierImpl();
24
25     CallableStatement cs = con.prepareCall("SELECT "+supplier.getColumn()+" FROM JAVADB.DATATYPE_MIXED_AL
26     cs.execute();
27 }
28 }
Cursor position: 697

Target Amazon RDS for PostgreSQL file: CallMethod2.java
14 private final String PASSWORD = "min_privs";
15
16
17 public CallMethod2(String conn_string) {
18     CONN_STRING = conn_string;
19 }
20
21 public void runExample() throws SQLException {
22     Connection con = DriverManager.getConnection(CONN_STRING, USER, PASSWORD);
23     Supplier supplier=new SupplierImpl();
24
25     CallableStatement cs = con.prepareCall("SELECT "+supplier.getColumn()+" FROM javadb.datatype_mixed_al
26     cs.execute();
27 }
28 }
29 }
```

4. Simpan kode aplikasi Anda yang telah dikonversi. Untuk informasi selengkapnya, lihat [Menyimpan kode aplikasi Anda yang telah dikonversi](#).

Aplikasi Java Anda mungkin menyertakan SQL kode yang berinteraksi dengan database sumber yang berbeda. Anda dapat bermigrasi ke Postgre SQL beberapa database sumber ini. Dalam hal ini, pastikan Anda tidak mengonversi SQL kode yang berinteraksi dengan database yang dikecualikan dari cakupan migrasi. Anda dapat mengecualikan file sumber aplikasi Java Anda dari lingkup konversi. Untuk melakukannya, kosongkan kotak centang untuk nama file yang ingin Anda keculikan dari lingkup konversi.

Setelah Anda mengubah cakupan konversi, AWS SCT masih menganalisis SQL kode semua file sumber aplikasi Java Anda. Kemudian, AWS SCT salin ke folder target semua file sumber yang Anda keculikan dari lingkup konversi. Operasi ini memungkinkan untuk membangun aplikasi Anda setelah Anda menyimpan file aplikasi yang dikonversi.

Menyimpan kode aplikasi Anda yang dikonversi dengan AWS SCT

Gunakan prosedur berikut untuk menyimpan kode aplikasi Anda yang telah dikonversi.

Untuk menyimpan kode aplikasi yang telah dikonversi

1. Perluas node Java di bawah Aplikasi di panel database target.
2. Pilih aplikasi yang dikonversi, dan pilih Simpan.
3. Masukkan path ke folder untuk menyimpan kode aplikasi yang dikonversi, dan pilih Pilih folder.

Jika aplikasi Java sumber Anda menggunakan MyBatis kerangka kerja, pastikan bahwa Anda memperbarui file konfigurasi Anda untuk bekerja dengan database baru Anda.

Mengelola proyek konversi aplikasi Java di AWS SCT

Anda dapat menambahkan beberapa proyek konversi aplikasi Java, memperbarui kode aplikasi dalam AWS SCT proyek, atau menghapus proyek konversi Java dari AWS SCT proyek Anda.

Untuk menambahkan proyek konversi aplikasi Java tambahan

1. Perluas node Applications di panel kiri.
2. Pilih node Java, dan buka menu konteks (klik kanan).
3. Pilih Aplikasi baru.
4. Masukkan informasi yang diperlukan untuk membuat proyek konversi aplikasi Java baru. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi Java](#).

Setelah Anda membuat perubahan dalam kode aplikasi sumber Anda, unggah ke dalam AWS SCT proyek.

Untuk mengunggah kode aplikasi yang diperbarui

1. Perluas node Java di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan diperbarui, dan buka menu konteks (klik kanan).
3. Pilih Refresh dan kemudian pilih Yes.

AWS SCT mengunggah kode aplikasi Anda dari file sumber dan menghapus hasil konversi. Untuk menyimpan perubahan kode yang Anda buat AWS SCT dan hasil konversi, buat proyek konversi Java baru.

Jika aplikasi Java sumber Anda menggunakan MyBatis kerangka kerja, AWS SCT gunakan file MyBatis konfigurasi untuk mengurai SQL kode Anda. Setelah Anda mengubah file ini, unggah ke dalam AWS SCT proyek.

Untuk mengedit jalur ke file MyBatis konfigurasi

1. Perluas node Java di bawah Aplikasi di panel kiri.
2. Pilih aplikasi Anda, lalu pilih Pengaturan.
3. Pilih Browse, lalu pilih file MyBatis konfigurasi.
4. Pilih Terapkan.
5. Di panel kiri, pilih aplikasi Anda, buka menu konteks (klik kanan), dan pilih Refresh.

Untuk menghapus proyek konversi aplikasi Java

1. Perluas node Java di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus dan kemudian pilih OK.

Membuat laporan penilaian konversi aplikasi Java di AWS SCT

Laporan penilaian konversi aplikasi Java memberikan informasi tentang mengonversi SQL kode yang disematkan dalam aplikasi Java Anda ke format yang kompatibel dengan basis data target Anda. Laporan penilaian memberikan detail konversi untuk semua titik SQL eksekusi dan semua file kode sumber. Laporan penilaian juga menyertakan item tindakan untuk SQL kode yang tidak AWS SCT dapat dikonversi.

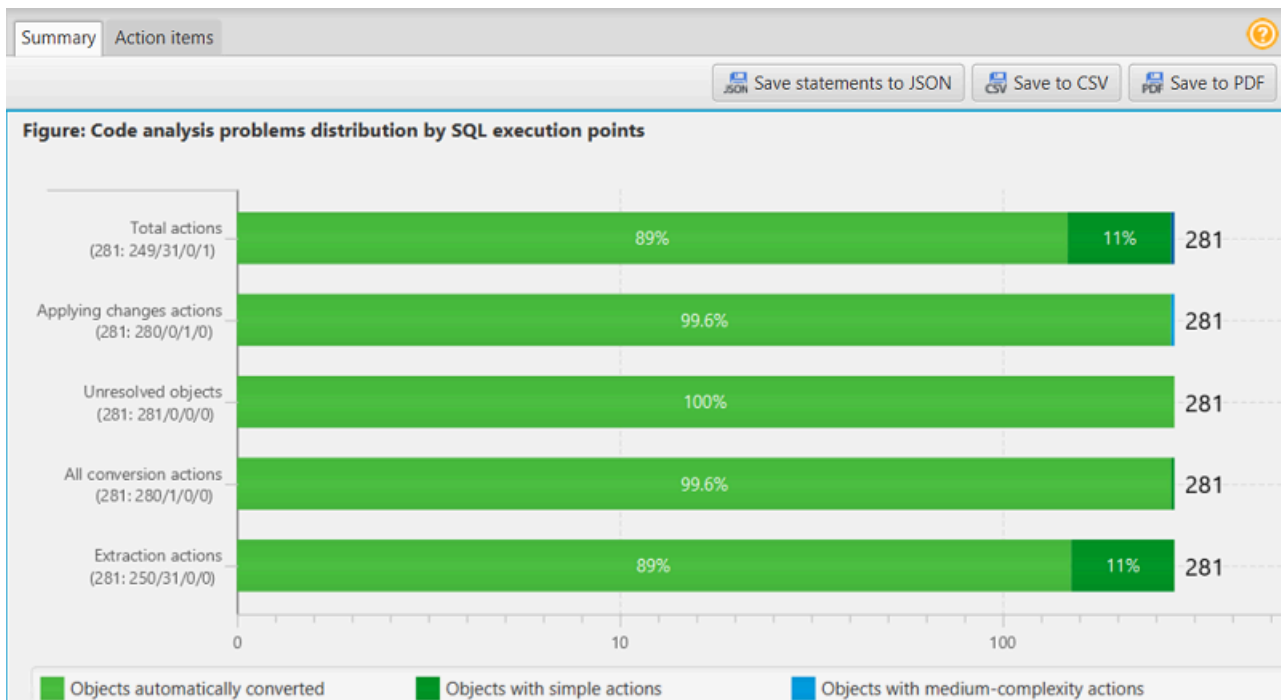
Gunakan prosedur berikut untuk membuat laporan penilaian konversi aplikasi Java.

Untuk membuat laporan penilaian konversi aplikasi Java

1. Perluas node Java di bawah Aplikasi di panel kiri.
2. Pilih aplikasi yang akan dikonversi, dan buka menu konteks (klik kanan).

3. Pilih Konversi.
4. Pada menu Lihat, pilih Tampilan laporan penilaian.
5. Tinjau tab Ringkasan.

Tab Ringkasan, ditampilkan berikut, menampilkan informasi ringkasan eksekutif dari laporan penilaian aplikasi Java. Ini menunjukkan hasil konversi untuk semua titik SQL eksekusi dan semua file kode sumber.



6. Pilih Simpan pernyataan JSON untuk menyimpan SQL kode yang diekstrak dari aplikasi Java Anda sebagai JSON file.
7. (Opsional) Simpan salinan lokal laporan sebagai PDF file atau file values (CSV) yang dipisahkan koma:
 - Pilih Simpan ke PDF kanan atas untuk menyimpan laporan sebagai PDF file.
PDFFile berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi aplikasi.
 - Pilih Simpan ke CSV kanan atas untuk menyimpan laporan sebagai CSV file.
CSVFile berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi SQL kode.

Mengonversi SQL kode dalam aplikasi Pro* C dengan AWS Schema Conversion Tool

Untuk SQL konversi Oracle ke Postgre, Anda dapat menggunakan AWS Schema Conversion Tool (AWS SCT) untuk mengonversi SQL kode yang disematkan ke dalam aplikasi Pro*C Anda. Konverter aplikasi Pro* C khusus ini memahami logika aplikasi. Ini mengumpulkan pernyataan yang terletak di objek aplikasi yang berbeda, seperti fungsi, parameter, variabel lokal, dan sebagainya.

Karena analisis mendalam ini, konverter SQL kode aplikasi Pro* C memberikan hasil konversi yang lebih baik dibandingkan dengan konverter generik.

Membuat proyek konversi aplikasi Pro* C di AWS SCT

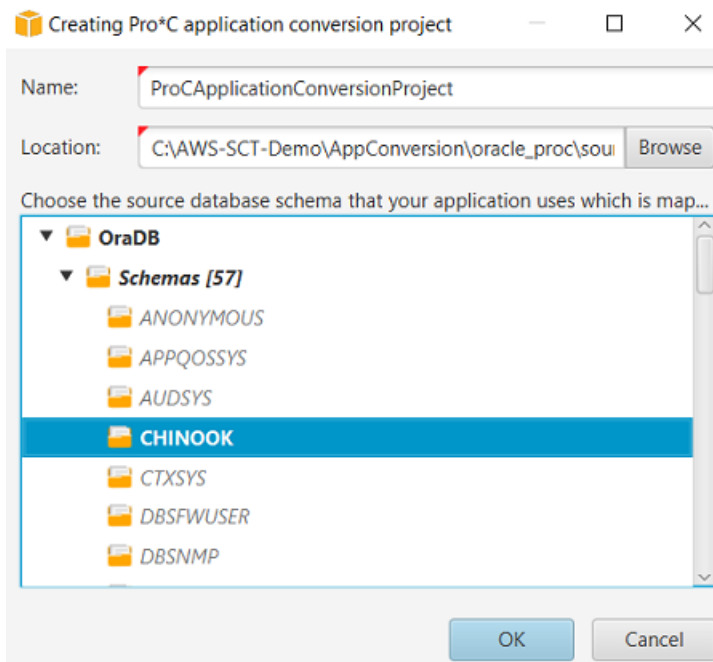
Anda dapat membuat proyek konversi aplikasi Pro* C hanya untuk mengonversi skema database Oracle ke skema database Postgre. Pastikan Anda menambahkan aturan pemetaan dalam proyek Anda yang menyertakan skema Oracle sumber dan database Postgre target. Untuk informasi selengkapnya, lihat [Memetakan tipe data di AWS Schema Conversion Tool](#).

Anda dapat menambahkan beberapa proyek konversi aplikasi dalam satu AWS SCT proyek. Gunakan prosedur berikut untuk membuat proyek konversi aplikasi Pro* C.

Untuk membuat proyek konversi aplikasi Pro*C

1. Buat proyek konversi database, dan tambahkan database Oracle sumber. Untuk informasi selengkapnya, silakan lihat [Memulai dan mengelola Proyek di AWS SCT](#) dan [Menambahkan server ke proyek di AWS SCT](#).
2. Tambahkan aturan pemetaan yang mencakup database Oracle sumber Anda dan database SQL Postgre target. Anda dapat menambahkan database Postgre target atau menggunakan platform SQL database SQL target Postgre virtual dalam aturan pemetaan. Untuk informasi selengkapnya, silakan lihat [Memetakan tipe data di AWS Schema Conversion Tool](#) dan [Pemetaan ke target virtual di AWS Schema Conversion Tool](#).
3. Pada menu Tampilan, pilih Tampilan utama.
4. Pada menu Applications, pilih aplikasi New Pro*C.

Kotak dialog Creating a Pro*C project konversi aplikasi muncul.



5. Untuk Nama, masukkan nama untuk proyek konversi aplikasi Pro*C Anda. Karena setiap proyek konversi skema database dapat memiliki satu atau lebih proyek konversi aplikasi anak, pilih nama yang masuk akal jika Anda menambahkan beberapa proyek.
6. Untuk Lokasi, masukkan lokasi kode sumber untuk aplikasi Anda.
7. Di pohon sumber, pilih skema yang digunakan aplikasi Anda. Pastikan bahwa skema ini adalah bagian dari aturan pemetaan. AWS SCT menyoroti skema yang merupakan bagian dari aturan pemetaan dalam huruf tebal.
8. Pilih OK untuk membuat proyek konversi aplikasi Pro* C Anda.
9. Temukan proyek konversi aplikasi Pro*C Anda di node Applications di panel kiri.

Mengonversi kode aplikasi Pro* C Anda SQL AWS SCT

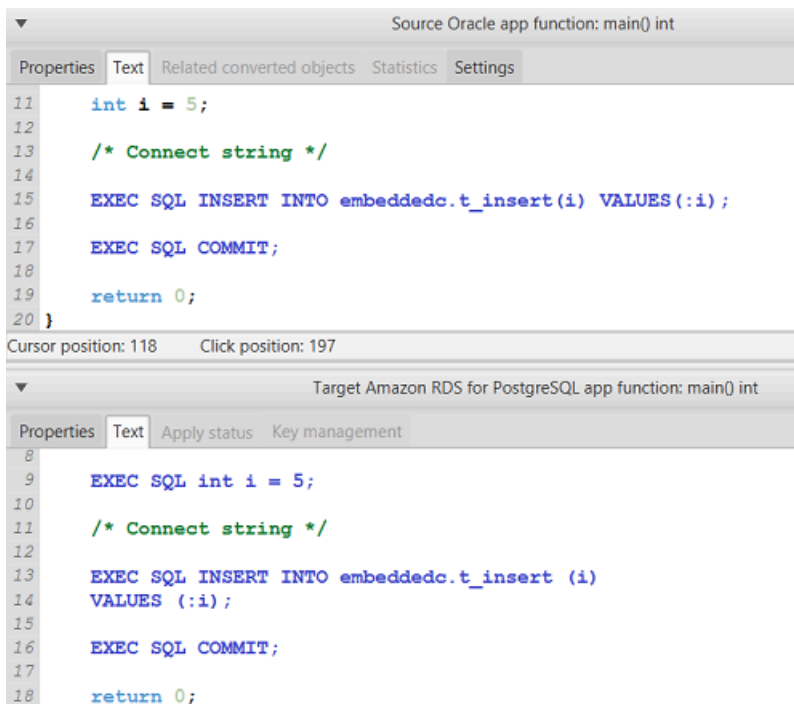
Setelah Anda menambahkan aplikasi Pro* C Anda ke AWS SCT proyek, konversi SQL kode dari aplikasi ini ke format yang kompatibel dengan platform database target Anda. Gunakan prosedur berikut untuk menganalisis dan mengonversi SQL kode yang disematkan dalam aplikasi Pro* C Anda di. AWS Schema Conversion Tool

Untuk mengonversi SQL kode Anda

1. Perluas node Pro*C di bawah Applications di panel kiri.
2. Pilih aplikasi yang akan dikonversi dan kemudian pilih Pengaturan.

- a. Untuk jalur file header Global, masukkan path ke file header yang digunakan proyek aplikasi Anda.
 - b. Pilih Menafsirkan semua variabel host yang belum terselesaikan untuk melihat semua variabel yang belum terselesaikan dalam kode yang dikonversi.
 - c. Pilih Gunakan fungsi konversi string dengan lebar tetap dari paket ekstensi untuk menggunakan fungsi paket ekstensi dalam kode yang dikonversi SQL. AWS SCT termasuk file paket ekstensi dalam proyek aplikasi Anda.
 - d. Pilih Transform SQL PL/blok anonim ke SQL panggilan mandiri atau fungsi tersimpan untuk membuat prosedur tersimpan dalam database target Anda untuk semua SQL PL/blok anonim. AWS SCT kemudian termasuk menjalankan prosedur yang disimpan ini dalam kode aplikasi yang dikonversi.
 - e. Pilih Gunakan alur kursor khusus untuk meningkatkan konversi kursor database Oracle.
3. Di panel kiri, pilih aplikasi yang akan dikonversi, dan buka menu konteks (klik kanan).
 4. Pilih Konversi. AWS SCT menganalisis file kode sumber Anda, menentukan logika aplikasi, dan memuat metadata kode ke dalam proyek. Metadata kode ini mencakup kelas Pro*C, objek, metode, variabel global, antarmuka, dan sebagainya.

Di panel database target, AWS SCT buat struktur folder yang mirip dengan proyek aplikasi sumber Anda. Di sini Anda dapat meninjau kode aplikasi yang dikonversi.



The screenshot displays two panels of code. The top panel, titled 'Source Oracle app function: main() int', shows the original Oracle PL/SQL code. The bottom panel, titled 'Target Amazon RDS for PostgreSQL app function: main() int', shows the converted PostgreSQL code. The conversion process has wrapped the original code in an EXEC SQL block.

```
Source Oracle app function: main() int
Properties Text Related converted objects Statistics Settings
11 int i = 5;
12
13 /* Connect string */
14
15 EXEC SQL INSERT INTO embeddedc.t_insert(i) VALUES (:i);
16
17 EXEC SQL COMMIT;
18
19 return 0;
20 }
Cursor position: 118 Click position: 197

Target Amazon RDS for PostgreSQL app function: main() int
Properties Text Apply status Key management
8
9 EXEC SQL int i = 5;
10
11 /* Connect string */
12
13 EXEC SQL INSERT INTO embeddedc.t_insert (i)
14 VALUES (:i);
15
16 EXEC SQL COMMIT;
17
18 return 0;
```

5. Simpan kode aplikasi Anda yang telah dikonversi. Untuk informasi selengkapnya, lihat [Mengedit dan menyimpan kode aplikasi Anda yang telah dikonversi](#).

Mengedit dan menyimpan kode aplikasi Anda yang dikonversi dengan AWS SCT

Anda dapat mengedit SQL pernyataan yang dikonversi dan digunakan AWS SCT untuk menyematkan kode yang diedit ini ke dalam kode aplikasi Pro*C yang dikonversi. Gunakan prosedur berikut untuk mengedit SQL kode konversi Anda.

Untuk mengedit SQL kode yang telah dikonversi

1. Perluas node Pro*C di bawah Applications di panel kiri.
2. Pilih aplikasi yang akan dikonversi, buka menu konteks (klik kanan), dan pilih Konversi.
3. Pada menu Lihat, pilih Tampilan laporan penilaian.
4. Pilih Simpan pernyataan CSV untuk menyimpan SQL kode yang diekstrak dari aplikasi Pro*C Anda sebagai file. CSV
5. Masukkan nama CSV file untuk menyimpan SQL kode yang diekstrak, dan pilih Simpan.
6. Edit SQL kode yang diekstraksi.
7. Pada menu Tampilan, pilih Tampilan utama.
8. Perluas node Pro*C di bawah Applications di panel database target.
9. Pilih aplikasi yang dikonversi, buka menu konteks (klik kanan), dan pilih Impor pernyataan dari CSV.
10. Pilih Ya, lalu pilih file dengan SQL kode yang diedit, dan pilih Buka.

AWS SCT memecah SQL pernyataan yang dikonversi menjadi beberapa bagian dan menempatkannya ke objek yang sesuai dari kode aplikasi sumber Anda. Gunakan prosedur berikut untuk menyimpan kode aplikasi Anda yang telah dikonversi.

Untuk menyimpan kode aplikasi yang telah dikonversi

1. Perluas node Pro*C di bawah Applications di panel database target.
2. Pilih aplikasi yang dikonversi, dan pilih Simpan.
3. Masukkan path ke folder untuk menyimpan kode aplikasi yang dikonversi, dan pilih Pilih folder.

Mengelola proyek konversi aplikasi Pro* C di AWS SCT

Anda dapat menambahkan beberapa proyek konversi aplikasi Pro*C, memperbarui kode aplikasi dalam AWS SCT proyek, atau menghapus proyek konversi Pro*C dari proyek Anda. AWS SCT

Untuk menambahkan proyek konversi aplikasi Pro* C tambahan

1. Perluas node Applications di panel kiri.
2. Pilih node Pro* C, dan buka menu konteks (klik kanan).
3. Pilih Aplikasi baru.
4. Masukkan informasi yang diperlukan untuk membuat proyek konversi aplikasi Pro*C baru. Untuk informasi selengkapnya, lihat [Membuat proyek konversi aplikasi Pro* C](#).

Setelah Anda membuat perubahan dalam kode aplikasi sumber Anda, unggah ke dalam AWS SCT proyek.

Untuk mengunggah kode aplikasi yang diperbarui

1. Perluas node Pro*C di bawah Applications di panel kiri.
2. Pilih aplikasi yang akan diperbarui, dan buka menu konteks (klik kanan).
3. Pilih Refresh dan kemudian pilih Yes.

AWS SCT mengunggah kode aplikasi Anda dari file sumber dan menghapus hasil konversi. Untuk menyimpan perubahan kode yang Anda buat AWS SCT dan hasil konversi, buat proyek konversi Pro*C baru.

Untuk menghapus proyek konversi aplikasi Pro* C

1. Perluas node Pro*C di bawah Applications di panel kiri.
2. Pilih aplikasi yang akan dihapus, dan buka menu konteks (klik kanan).
3. Pilih Hapus dan kemudian pilih OK.

Membuat laporan penilaian konversi aplikasi Pro*C di AWS SCT

Laporan penilaian konversi aplikasi Pro*C memberikan informasi tentang mengonversi SQL kode yang disematkan dalam aplikasi Pro*C Anda ke format yang kompatibel dengan basis data target

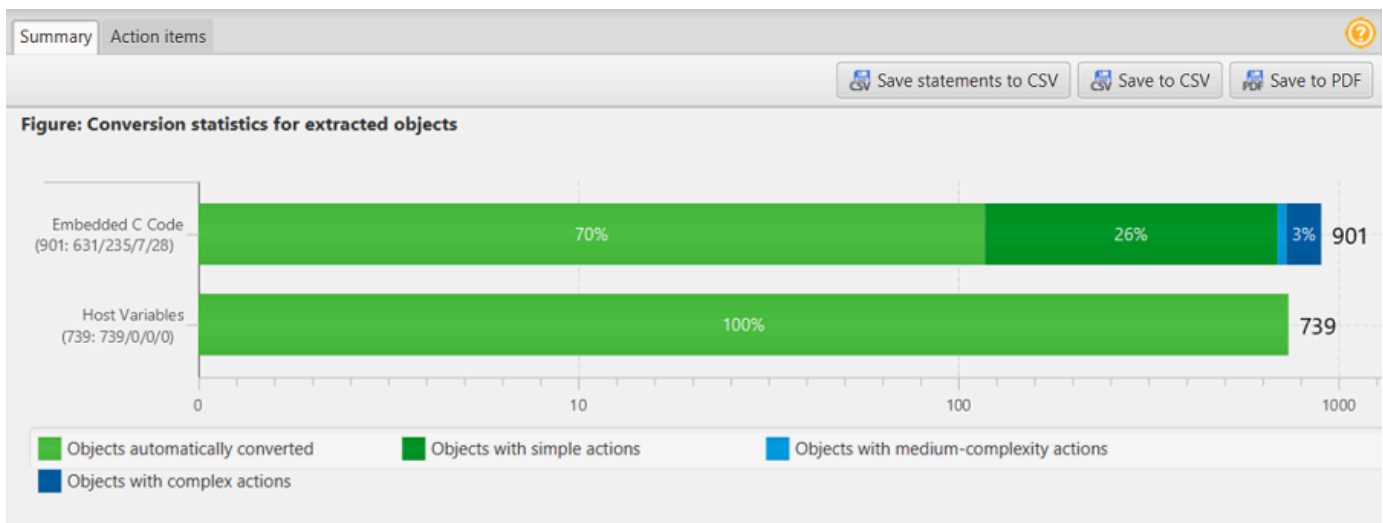
Anda. Laporan penilaian memberikan detail konversi untuk semua titik SQL eksekusi dan semua file kode sumber. Laporan penilaian juga menyertakan item tindakan untuk SQL kode yang tidak AWS SCT dapat dikonversi.

Gunakan prosedur berikut untuk membuat laporan penilaian konversi aplikasi Pro*C.

Untuk membuat laporan penilaian konversi aplikasi Pro*C

1. Perluas node Pro*C di bawah Applications di panel kiri.
2. Pilih aplikasi yang akan dikonversi, dan buka menu konteks (klik kanan).
3. Pilih Konversi.
4. Pada menu Lihat, pilih Tampilan laporan penilaian.
5. Tinjau tab Ringkasan.

Tab Ringkasan, ditampilkan berikut, menampilkan informasi ringkasan eksekutif dari laporan penilaian aplikasi Pro*C. Ini menunjukkan hasil konversi untuk semua titik SQL eksekusi dan semua file kode sumber.



6. Pilih Simpan pernyataan CSV untuk menyimpan SQL kode yang diekstrak dari aplikasi Pro*C Anda sebagai file values (,) yang dipisahkan koma. CSV
7. (Opsional) Simpan salinan lokal laporan sebagai PDF file atau file values (CSV) yang dipisahkan koma:
 - Pilih Simpan ke PDF kanan atas untuk menyimpan laporan sebagai PDF file.
PDFFile berisi ringkasan eksekutif, item tindakan, dan rekomendasi untuk konversi aplikasi.
 - Pilih Simpan ke CSV kanan atas untuk menyimpan laporan sebagai CSV file.

CSVFile berisi item tindakan, tindakan yang direkomendasikan, dan perkiraan kompleksitas upaya manual yang diperlukan untuk mengonversi SQL kode.

Menggunakan paket ekstensi dengan AWS Schema Conversion Tool

Paket AWS SCT ekstensi adalah modul add-on yang mengemulasi fungsi yang ada dalam database sumber yang diperlukan saat mengonversi objek ke database target. Sebelum Anda dapat menginstal paket AWS SCT ekstensi, Anda mengonversi skema database Anda.

Setiap paket AWS SCT ekstensi mencakup komponen-komponen berikut:

- Skema DB — Termasuk SQL fungsi, prosedur, dan tabel untuk meniru pemrosesan transaksi online tertentu (OLTP) dan pemrosesan analitik online (OLAP) objek database seperti urutan. Juga, mengemulasi tidak didukung built-in-functions dari database sumber. Nama skema ini memiliki format sebagai berikut:aws_*database_engine_name*_ext.
- AWS Lambda fungsi (untuk OLTP database tertentu) - Termasuk AWS Lambda fungsi yang meniru fungsionalitas database yang kompleks, seperti penjadwalan pekerjaan dan mengirim email.
- Pustaka khusus untuk OLAP database - Termasuk sekumpulan pustaka Java dan Python yang dapat Anda gunakan untuk memigrasi Microsoft SQL Server Integration Services (SSIS) mengekstrak, mengubah, dan memuat () skrip ke atau. ETL AWS Glue AWS Glue Studio

Pustaka Java mencakup modul-modul berikut:

- spark-excel_2.11-0.13.1.jar— Untuk meniru fungsionalitas sumber Excel dan komponen target.
- spark-xml_2.11-0.9.0.jar,poi-ooxml-schemas-4.1.2.jar, dan xmlbeans-3.1.0.jar — Untuk meniru fungsionalitas komponen XML sumber.

Pustaka Python mencakup modul-modul berikut:

- sct_utils.py— Untuk meniru tipe data sumber dan menyiapkan parameter untuk kueri SparkSQL.
- ssis_datetime.py— Untuk meniru fungsi bawaan tanggal dan waktu.
- ssis_null.py— Untuk meniru ISNULL dan fungsi REPLACENULL bawaan.
- ssis_string.py- Untuk meniru fungsi bawaan string.

Untuk informasi lebih lanjut tentang perpustakaan ini, lihat[Menggunakan pustaka khusus untuk paket AWS SCT ekstensi](#).

Anda dapat menerapkan paket AWS SCT ekstensi dengan dua cara:

- AWS SCT dapat secara otomatis menerapkan paket ekstensi saat Anda menerapkan skrip basis data target dengan memilih Terapkan ke database dari menu konteks. AWS SCT menerapkan paket ekstensi sebelum menerapkan semua objek skema lainnya.
- Untuk menerapkan paket ekstensi secara manual, pilih database target dan kemudian pilih Terapkan paket ekstensi untuk dari menu konteks (klik kanan). Dalam kebanyakan situasi, aplikasi otomatis sudah cukup. Namun, Anda mungkin ingin menerapkan paket secara manual jika tidak sengaja dihapus.

Setiap kali Anda menerapkan paket AWS SCT ekstensi ke penyimpanan data target, komponen akan ditimpa, dan AWS SCT menampilkan pemberitahuan tentang hal ini. Untuk mematikan notifikasi ini, pilih Pengaturan, Pengaturan global, Pemberitahuan, lalu pilih Sembunyikan peringatan penggantian paket ekstensi.

Untuk konversi dari Microsoft SQL Server ke PostgreSQL, Anda dapat menggunakan paket ekstensi SQL Server ke Postgre SQL di. AWS SCT Paket ekstensi ini mengemulasi SQL Server Agent dan SQL Server Database Mail. Untuk informasi selengkapnya, silakan lihat [Meniru Agen SQL Server di Postgre SQL dengan paket ekstensi](#) dan [Meniru SQL Server Database Mail di Postgre SQL dengan paket ekstensi](#).

Berikut ini, Anda dapat menemukan informasi lebih lanjut tentang bekerja dengan paket AWS SCT ekstensi.

Topik

- [Izin untuk menggunakan paket AWS SCT ekstensi](#)
- [Menggunakan skema paket ekstensi](#)
- [Menggunakan pustaka khusus untuk paket AWS SCT ekstensi](#)
- [Menggunakan AWS Lambda fungsi dari paket AWS SCT ekstensi](#)
- [Mengkonfigurasi fungsi untuk paket AWS SCT ekstensi](#)

Izin untuk menggunakan paket AWS SCT ekstensi

Paket AWS SCT ekstensi untuk Amazon Aurora mengemulasi pengiriman email, penjadwalan pekerjaan, antrian, dan operasi lainnya menggunakan fungsi. AWS Lambda Saat Anda menerapkan paket AWS SCT ekstensi ke basis data Aurora target Anda, AWS SCT buat peran new AWS Identity

and Access Management (IAM) dan kebijakan IAM inline. Selanjutnya, AWS SCT buat fungsi Lambda baru, dan konfigurasi cluster Aurora DB Anda untuk koneksi keluar. AWS Lambda Untuk menjalankan operasi ini, pastikan Anda memberikan izin yang diperlukan berikut kepada IAM pengguna Anda:

- `iam:CreateRole`— untuk membuat IAM peran baru untuk AWS akun Anda.
- `iam:CreatePolicy`— untuk membuat IAM kebijakan baru untuk AWS akun Anda.
- `iam:AttachRolePolicy`— untuk melampirkan kebijakan yang ditentukan ke IAM peran Anda.
- `iam:PutRolePolicy`— untuk memperbarui dokumen kebijakan inline yang disematkan dalam IAM peran Anda.
- `iam:PassRole`— untuk meneruskan IAM peran yang ditentukan ke mesin aturan.
- `iam:TagRole`— untuk menambahkan tag ke IAM peran.
- `iam:TagPolicy`— untuk menambahkan tag ke IAM kebijakan.
- `lambda:ListFunctions`— untuk melihat daftar fungsi Lambda Anda.
- `lambda:ListTags`— untuk melihat daftar tag fungsi Lambda Anda.
- `lambda:CreateFunction`— untuk membuat fungsi Lambda baru.
- `rds:AddRoleToDBCluster`— untuk mengaitkan IAM peran dengan cluster Aurora DB Anda.

Paket AWS SCT ekstensi untuk Amazon Redshift mengemulasi fungsi basis gudang data sumber yang diperlukan saat menerapkan objek yang dikonversi ke Amazon Redshift. Sebelum Anda menerapkan kode yang dikonversi ke Amazon Redshift, Anda harus menerapkan paket ekstensi untuk Amazon Redshift. Untuk melakukannya, sertakan `iam:SimulatePrincipalPolicy` tindakan dalam IAM kebijakan Anda.

AWS SCT menggunakan Simulator IAM Kebijakan untuk memeriksa izin yang diperlukan untuk menginstal paket ekstensi Amazon Redshift. Simulator IAM Kebijakan dapat menampilkan pesan kesalahan meskipun Anda telah mengonfigurasi IAM pengguna dengan benar. Ini adalah masalah yang diketahui dari Simulator IAM Kebijakan. Selain itu, Simulator IAM Kebijakan akan menampilkan pesan galat saat Anda tidak memiliki `iam:SimulatePrincipalPolicy` tindakan dalam IAM kebijakan Anda. Dalam kasus ini, Anda dapat mengabaikan pesan kesalahan dan menerapkan paket ekstensi menggunakan wizard paket ekstensi. Untuk informasi selengkapnya, lihat [Menerapkan paket ekstensi](#).

Menggunakan skema paket ekstensi

Saat Anda mengonversi database atau skema gudang data, AWS SCT tambahkan skema tambahan ke basis data target Anda. Skema ini mengimplementasikan fungsi SQL sistem dari database sumber yang diperlukan saat menulis skema yang dikonversi ke database target Anda. Skema tambahan ini disebut skema paket ekstensi.

Skema paket ekstensi untuk OLTP database diberi nama sesuai dengan basis data sumber sebagai berikut:

- SQLServer Microsoft: `AWS_SQLSERVER_EXT`
- SayaSQL: `AWS_MYSQL_EXT`
- Oracle: `AWS_ORACLE_EXT`
- PostgreSQL: `AWS_POSTGRESQL_EXT`

Skema paket ekstensi untuk aplikasi gudang OLAP data diberi nama sesuai dengan penyimpanan data sumber sebagai berikut:

- Greenplum: `AWS_GREENPLUM_EXT`
- SQLServer Microsoft: `AWS_SQLSERVER_EXT`
- Netezza: `AWS_NETEZZA_EXT`
- Oracle: `AWS_ORACLE_EXT`
- Teradata: `AWS_TERADATA_EXT`
- Vertika: `AWS_VERTICA_EXT`

Menggunakan pustaka khusus untuk paket AWS SCT ekstensi

Dalam beberapa kasus, tidak AWS SCT dapat mengonversi fitur database sumber ke fitur yang setara dalam database target Anda. Paket AWS SCT ekstensi yang relevan berisi pustaka khusus yang meniru beberapa fungsionalitas basis data sumber di database target Anda.

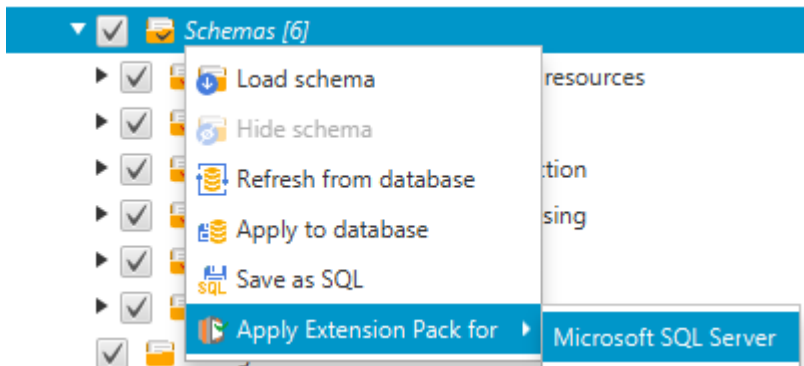
Jika Anda mengonversi database transaksional, lihat. [Menggunakan AWS Lambda fungsi dari paket AWS SCT ekstensi](#)

Menerapkan paket ekstensi

Anda dapat menerapkan paket AWS SCT ekstensi menggunakan wizard paket ekstensi atau ketika Anda menerapkan kode yang dikonversi ke database target Anda.

Untuk menerapkan paket ekstensi menggunakan wizard paket ekstensi

1. Di AWS Schema Conversion Tool, di pohon database target, buka menu konteks (klik kanan), pilih Terapkan paket ekstensi untuk, lalu pilih platform basis data sumber Anda.



Wizard paket ekstensi muncul.

2. Baca halaman Selamat Datang, lalu pilih Berikutnya.
3. Pada halaman pengaturan AWS profil, lakukan hal berikut:
 - Jika Anda menginstal ulang skema paket ekstensi saja, pilih Lewati langkah ini untuk saat ini, lalu pilih Berikutnya. Opsi Lewati langkah ini untuk saat ini hanya tersedia untuk database pemrosesan transaksi online (OLTP).
 - Jika Anda mengunggah perpustakaan baru, berikan kredensialnya untuk terhubung ke perpustakaan Anda. Akun AWS Gunakan langkah ini hanya ketika Anda mengonversi OLAP database atau ETL skrip. Anda dapat menggunakan kredensi AWS Command Line Interface (AWS CLI) Anda jika Anda telah menginstal. AWS CLI Anda juga dapat menggunakan kredensial yang sebelumnya Anda simpan di profil di pengaturan aplikasi global dan terkait dengan proyek. Jika perlu, pilih Navigasikan ke pengaturan global untuk mengonfigurasi kaitkan profil yang berbeda dengan AWS SCT proyek Anda. Untuk informasi selengkapnya, lihat [Mengelola Profil di AWS Schema Conversion Tool](#).
4. Jika Anda mengunggah perpustakaan baru, maka pilih Saya perlu mengunggah perpustakaan di halaman unggah Perpustakaan. Gunakan langkah ini hanya ketika Anda mengonversi OLAP database atau ETL skrip. Selanjutnya, berikan jalur Amazon S3, lalu pilih Upload library ke S3.

Jika Anda sudah mengunggah perpustakaan, lalu pilih Saya sudah mengunggah perpustakaan, gunakan bucket S3 saya yang ada di halaman unggah Perpustakaan. Selanjutnya, berikan jalur Amazon S3.

Setelah selesai, pilih Berikutnya.

5. Pada halaman Emulasi fungsi, pilih Buat paket ekstensi. Pesan muncul dengan status operasi paket ekstensi.

Setelah selesai, pilih Selesai.

Untuk menerapkan paket ekstensi saat menerapkan kode yang dikonversi

1. Tentukan bucket Amazon S3 di profil AWS layanan Anda. Gunakan langkah ini hanya ketika Anda mengonversi OLAP database atau ETL skrip. Untuk informasi selengkapnya, lihat [Mengelola Profil di AWS Schema Conversion Tool](#).

Pastikan kebijakan bucket Amazon S3 Anda menyertakan izin berikut:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["s3:ListBucket"],
      "Resource": ["*"]
    },
    {
      "Effect": "Allow",
      "Action": ["s3:PutObject"],
      "Resource": ["*"]
    },
    {
      "Effect": "Allow",
      "Action": ["iam:SimulatePrincipalPolicy"],
      "Resource": ["*"]
    },
    {
      "Effect": "Allow",
      "Action": ["iam:GetUser"],
      "Resource": ["arn:aws:iam::111122223333:user/DataExtractionAgentName"]
    }
  ]
}
```

```
    }  
  ]  
}
```

Pada contoh sebelumnya, ganti `111122223333:user/DataExtractionAgentName` dengan nama IAM pengguna Anda.

2. Konversikan skema gudang data sumber Anda. Untuk informasi selengkapnya, lihat [Mengonversi skema gudang data](#).
3. Di panel kanan, pilih skema yang dikonversi.
4. Buka menu konteks (klik kanan) untuk elemen skema, lalu pilih Terapkan ke database.
5. AWS SCT menghasilkan paket ekstensi dengan komponen yang diperlukan dan menambahkan `aws_database_engine_name_ext` skema di pohon target. Selanjutnya, AWS SCT terapkan kode yang dikonversi dan skema paket ekstensi ke gudang data target Anda.

Saat Anda menggunakan kombinasi Amazon Redshift dan AWS Glue sebagai platform basis data target Anda, AWS SCT tambahkan skema tambahan dalam paket ekstensi.

Menggunakan AWS Lambda fungsi dari paket AWS SCT ekstensi

AWS SCT menyediakan paket ekstensi yang berisi fungsi Lambda untuk email, penjadwalan pekerjaan, dan fitur lain untuk database yang dihosting di Amazon. EC2

Menggunakan AWS Lambda fungsi untuk meniru fungsionalitas database

Dalam beberapa kasus, fitur database tidak dapat dikonversi ke RDS fitur Amazon yang setara. Misalnya, Oracle mengirim panggilan email yang digunakan `UTL_SMTP`, dan Microsoft SQL Server dapat menggunakan penjadwal pekerjaan. Jika Anda meng-host dan mengelola sendiri database di AmazonEC2, Anda dapat meniru fitur-fitur ini dengan mengganti AWS layanan untuk mereka.

Wisaya paket AWS SCT ekstensi membantu Anda menginstal, membuat, dan mengonfigurasi fungsi Lambda untuk meniru email, penjadwalan pekerjaan, dan fitur lainnya.

Menerapkan paket ekstensi untuk mendukung fungsi Lambda

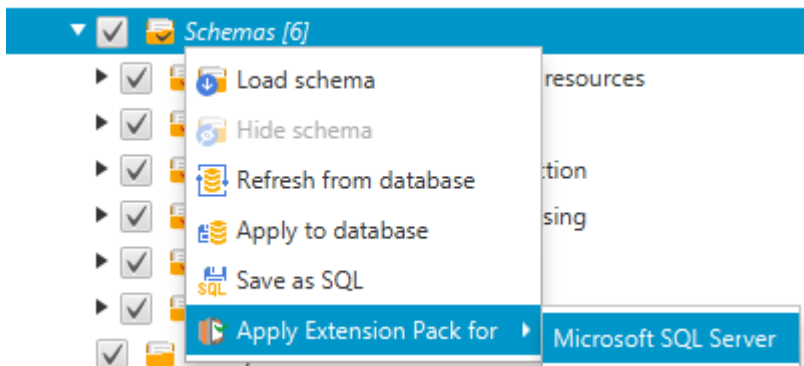
Anda dapat menerapkan paket ekstensi untuk mendukung fungsi Lambda menggunakan wizard paket ekstensi atau saat Anda menerapkan kode yang dikonversi ke basis data target Anda.

⚠ Important

Fitur emulasi AWS layanan hanya didukung untuk database yang diinstal dan dikelola sendiri di Amazon. EC2 Jangan menginstal fitur emulasi layanan jika database target Anda ada di instans Amazon RDS DB.

Untuk menerapkan paket ekstensi menggunakan wizard paket ekstensi

1. Di AWS Schema Conversion Tool, di pohon database target, buka menu konteks (klik kanan), pilih Terapkan paket ekstensi untuk, lalu pilih platform basis data sumber Anda.



Wizard paket ekstensi muncul.

2. Baca halaman Selamat Datang, lalu pilih Berikutnya.
3. Pada halaman pengaturan AWS profil, lakukan hal berikut:
 - Jika Anda menginstal ulang skema paket ekstensi saja, pilih Lewati langkah ini untuk saat ini, lalu pilih Berikutnya.
 - Jika Anda menginstal AWS layanan, berikan kredensi untuk terhubung ke layanan Anda. Akun AWS Anda dapat menggunakan AWS CLI kredensi Anda jika Anda telah menginstal. AWS CLI Anda juga dapat menggunakan kredensial yang sebelumnya Anda simpan di profil di pengaturan aplikasi global dan terkait dengan proyek. Jika perlu, pilih Navigasikan ke Pengaturan Proyek untuk mengaitkan profil yang berbeda dengan proyek. Jika perlu, pilih Pengaturan Global untuk membuat profil baru. Untuk informasi selengkapnya, lihat [Mengelola Profil di AWS Schema Conversion Tool](#).
4. Pada halaman Layanan Pengiriman Email, lakukan hal berikut:
 - Jika Anda menginstal ulang skema paket ekstensi saja, pilih Lewati langkah ini untuk saat ini, lalu pilih Berikutnya.

- Jika Anda menginstal AWS layanan dan Anda memiliki fungsi Lambda yang ada, Anda dapat menyediakannya. Jika tidak, wizard membuatnya untuk Anda. Setelah selesai, pilih Berikutnya.
5. Pada halaman Job Emulation Service, lakukan hal berikut:
 - Jika Anda menginstal ulang skema paket ekstensi saja, pilih Lewati langkah ini untuk saat ini, lalu pilih Berikutnya.
 - Jika Anda menginstal AWS layanan dan Anda memiliki fungsi Lambda yang ada, Anda dapat menyediakannya. Jika tidak, wizard membuatnya untuk Anda. Setelah selesai, pilih Berikutnya.
 6. Pada halaman Emulasi fungsi, pilih Buat paket ekstensi. Pesan muncul dengan status operasi paket ekstensi.

Setelah selesai, pilih Selesai.

Note

Untuk memperbarui paket ekstensi dan menimpa komponen paket ekstensi lama, pastikan Anda menggunakan versi terbaru. AWS SCT Untuk informasi selengkapnya, lihat [Instalasi dan Konfigurasi AWS Schema Conversion Tool](#).

Mengkonfigurasi fungsi untuk paket AWS SCT ekstensi

Paket ekstensi berisi fungsi yang harus Anda konfigurasi sebelum digunakan. Konstanta `CONVERSION_LANG` mendefinisikan bahasa yang digunakan paket layanan. Fungsinya tersedia untuk bahasa Inggris dan Jerman.

Untuk mengatur bahasa ke bahasa Inggris atau Jerman, buat perubahan berikut dalam kode fungsi. Temukan deklarasi konstan berikut:

```
CONVERSION_LANG CONSTANT VARCHAR := '';
```

Untuk mengatur `CONVERSION_LANG` ke bahasa Inggris, ubah baris menjadi berikut:

```
CONVERSION_LANG CONSTANT VARCHAR := 'English';
```

Untuk mengatur `CONVERSION_LANG` ke bahasa Inggris, ubah baris menjadi berikut:

```
CONVERSION_LANG CONSTANT VARCHAR := 'Deutsch';
```

Tetapkan pengaturan ini untuk fungsi-fungsi berikut:

- `aws_sqlserver_ext.conv_datetime_to_string`
- `aws_sqlserver_ext.conv_date_to_string`
- `aws_sqlserver_ext.conv_string_to_date`
- `aws_sqlserver_ext.conv_string_to_datetime`
- `aws_sqlserver_ext.conv_string_to_datetime`
- `aws_sqlserver_ext.parse_to_date`
- `aws_sqlserver_ext.parse_to_datetime`
- `aws_sqlserver_ext.parse_to_time`

Praktik terbaik untuk menggunakan AWS Schema Conversion Tool

Temukan informasi tentang praktik terbaik dan opsi untuk menggunakan AWS Schema Conversion Tool (AWS SCT).

Mengkonfigurasi memori tambahan

Untuk mengonversi skema database besar, seperti database dengan 3.500 prosedur tersimpan, Anda dapat mengkonfigurasi jumlah memori yang tersedia untuk file. AWS Schema Conversion Tool

Untuk memodifikasi jumlah memori yang AWS SCT dikonsumsi

1. Pada menu Pengaturan, pilih Pengaturan global, lalu pilih JVMopsi.
2. Pilih Edit file konfigurasi dan pilih editor teks untuk membuka file konfigurasi.
3. Edit `JavaOptions` bagian untuk mengatur memori minimum dan maksimum yang tersedia. Contoh berikut menetapkan minimum untuk empat GB dan maksimum untuk 40 GB.

```
[JavaOptions]
-Xmx40960M
-Xms4096M
```

Kami menyarankan Anda mengatur memori minimum yang tersedia untuk setidaknya empat GB.

4. Simpan file konfigurasi, pilih OK, dan mulai ulang AWS SCT untuk menerapkan perubahan.

Mengkonfigurasi folder proyek default

AWS SCT menggunakan folder proyek untuk menyimpan file proyek, menyimpan laporan penilaian, dan menyimpan kode yang dikonversi. Secara default, AWS SCT menyimpan semua file di folder aplikasi. Anda dapat menentukan folder lain sebagai folder proyek default.

Untuk mengubah folder proyek default

1. Pada menu Pengaturan, pilih Pengaturan global, lalu pilih Jalur file.
2. Untuk jalur file proyek Default, masukkan path ke folder proyek default.

3. Pilih Berlakukan, lalu pilih OKE.

Meningkatkan kecepatan migrasi data

Untuk memigrasikan kumpulan data besar, seperti kumpulan tabel dengan lebih dari 1 TB data, Anda mungkin ingin meningkatkan kecepatan migrasi. Saat Anda menggunakan agen ekstraksi data, kecepatan migrasi data bergantung pada berbagai faktor. Faktor-faktor ini termasuk jumlah irisan di kluster Amazon Redshift target Anda, ukuran file chunk dalam tugas migrasi Anda, tersedia RAM di PC tempat Anda menjalankan agen ekstraksi data, dan sebagainya.

Untuk meningkatkan kecepatan migrasi data, sebaiknya jalankan beberapa sesi uji migrasi dengan kumpulan data kecil dari data produksi Anda. Selain itu, kami menyarankan Anda menjalankan agen ekstraksi data Anda di PC dengan ukuran minimal 500 GB. SSD Selama sesi pengujian ini, ubah parameter migrasi yang berbeda memantau pemanfaatan disk Anda untuk mengetahui konfigurasi yang memastikan kecepatan migrasi data maksimum. Kemudian, gunakan konfigurasi ini untuk memigrasikan seluruh kumpulan data Anda.

Meningkatkan informasi pencatatan

Anda dapat meningkatkan informasi pencatatan yang dihasilkan AWS SCT saat mengonversi basis data, skrip, dan aplikasi Anda. SQL Meskipun meningkatkan informasi logging dapat memperlambat konversi, perubahan dapat membantu Anda memberikan informasi yang kuat ke AWS Support jika terjadi kesalahan.

AWS SCT menyimpan log di lingkungan lokal Anda. Anda dapat melihat file log ini dan membagikannya dengan AWS Support atau AWS SCT pengembang untuk pemecahan masalah.

Untuk mengubah setelan logging

1. Pada menu Pengaturan, pilih Pengaturan global, lalu pilih Logging.
2. Untuk jalur folder Log, masukkan folder untuk menyimpan log dari antarmuka pengguna.
3. Untuk jalur folder log konsol, masukkan folder untuk menyimpan log antarmuka baris AWS SCT perintah (CLI).
4. Untuk Ukuran file log maksimum (MB), masukkan ukuran, dalam MB, dari satu file log. Setelah file Anda mencapai batas ini, AWS SCT buat file log baru.
5. Untuk Jumlah maksimum file log, masukkan jumlah file log yang akan disimpan. Setelah jumlah file log di folder mencapai batas ini, AWS SCT hapus file log tertua.

6. Untuk jalur unduhan log Extractors, masukkan folder untuk menyimpan log agen ekstraksi data.
7. Untuk jalur log ekstraktor Cassandra, masukkan folder untuk menyimpan log agen ekstraksi data Apache Cassandra.
8. Pilih Minta jalur sebelum memuat untuk memastikan bahwa AWS SCT menanyakan tempat menyimpan log setiap kali Anda menggunakan agen ekstraksi data.
9. Untuk mode Debug, pilih True. Gunakan opsi ini untuk mencatat informasi tambahan ketika AWS SCT log standar tidak menyertakan masalah apa pun.
10. Pilih modul aplikasi utama untuk meningkatkan informasi pencatatan. Anda dapat meningkatkan informasi pencatatan untuk modul aplikasi berikut:
 - Umum
 - Loader
 - Parser
 - Printer
 - Penyelesai
 - Telemetri
 - Konverter
 - Jenis pemetaan
 - Antarmuka pengguna
 - Pengendali
 - Bandingkan skema
 - Pusat data klon
 - Penganalisis aplikasi

Untuk setiap modul aplikasi sebelumnya, pilih salah satu level logging berikut:

- Jejak — Informasi paling rinci.
- Debug — Informasi terperinci tentang aliran melalui sistem.
- Info — Acara runtime, seperti startup atau shutdown.
- Peringatan — Penggunaan situasi runtime lain yang APIs tidak digunakan lagi dan API buruk yang tidak diinginkan atau tidak terduga.
- Kesalahan — Kesalahan runtime atau kondisi tak terduga.
- Kritis — Kesalahan yang menyebabkan aplikasi dimatikan.

- **Wajib** — Tingkat kesalahan setinggi mungkin.

Secara default, setelah Anda mengaktifkan mode Debug, AWS SCT menetapkan tingkat pencatatan Info untuk semua modul aplikasi.

Misalnya, untuk membantu area masalah utama selama konversi, setel Parser, pemetaan Jenis, dan antarmuka Pengguna ke Trace.

Jika informasi menjadi terlalu bertele-tele untuk sistem file tempat log streaming, ubah ke lokasi dengan ruang yang cukup untuk menangkap log.

Untuk mengirimkan log ke AWS Support, buka direktori tempat log disimpan, dan kompres semua file menjadi satu file.zip yang dapat dikelola. Kemudian unggah file.zip dengan kasus dukungan Anda. Ketika analisis awal selesai dan pengembangan berkelanjutan dilanjutkan, kembalikan mode Debug ke false untuk menghilangkan logging verbose. Kemudian tingkatkan kecepatan konversi.

Tip

Untuk mengelola ukuran log dan merampingkan masalah pelaporan, hapus log atau pindahkan ke lokasi lain setelah konversi berhasil. Melakukan tugas ini memastikan bahwa hanya kesalahan dan informasi yang relevan yang dikirimkan ke AWS Support dan menjaga sistem file log agar tidak terisi.

Memecahkan masalah dengan AWS Schema Conversion Tool

Setelah itu, Anda dapat menemukan informasi tentang pemecahan masalah dengan AWS Schema Conversion Tool (AWS SCT).

Tidak dapat memuat objek dari database sumber Oracle

Ketika Anda mencoba untuk memuat skema dari database Oracle, Anda mungkin menemukan salah satu kesalahan berikut.

```
Cannot load objects tree.
```

```
ORA-00942: table or view does not exist
```

Kesalahan ini terjadi karena pengguna yang ID Anda gunakan untuk terhubung ke database Oracle tidak memiliki izin yang cukup untuk membaca skema, seperti yang dipersyaratkan oleh AWS SCT.

Anda dapat menyelesaikan masalah dengan memberikan izin pengguna dan juga `select_catalog_role` izin ke kamus apa pun di database. Izin ini menyediakan akses hanya-baca ke tampilan dan tabel sistem yang diperlukan oleh AWS SCT. Contoh berikut membuat ID pengguna bernama `min_privs` dan memberikan pengguna dengan ID ini izin minimum yang diperlukan untuk mengkonversi skema dari database sumber Oracle.

```
create user min_privs identified by min_privs;  
grant connect to min_privs;  
grant select_catalog_role to min_privs;  
grant select any dictionary to min_privs;
```

Pesan peringatan laporan penilaian

Untuk menilai kompleksitas konversi ke mesin database lain, AWS SCT memerlukan akses ke objek dalam database sumber Anda. Ketika AWS SCT mengalami masalah selama pemindaian dan tidak dapat melakukan penilaian, pesan peringatan dikeluarkan. Pesan ini menunjukkan bahwa persentase konversi keseluruhan berkurang. Berikut adalah alasan mengapa AWS SCT mungkin mengalami masalah selama pemindaian:

- Pengguna database Anda tidak memiliki akses ke semua objek yang dibutuhkan. Untuk informasi selengkapnya tentang izin keamanan yang AWS SCT diperlukan dan hak istimewa untuk database Anda, lihat [Menghubungkan ke database sumber dengan AWS Schema Conversion Tool](#) bagian basis data sumber yang sesuai dalam panduan ini.
- Objek yang dikutip dalam skema tidak lagi ada dalam database. Untuk membantu menyelesaikan masalah, Anda dapat terhubung dengan SYSDBA izin dan memeriksa apakah objek ada dalam database.
- SCT sedang mencoba menilai objek yang dienkripsi.

CLI Referensi untuk AWS Schema Conversion Tool

Bagian ini menjelaskan cara memulai dengan antarmuka baris perintah AWS SCT (CLI). Juga, bagian ini memberikan informasi tentang perintah utama dan mode penggunaan. Untuk referensi lengkap tentang AWS SCT CLI perintah, lihat [Bahan referensi](#).

Topik

- [Prasyarat untuk menggunakan antarmuka baris perintah AWS SCT](#)
- [AWS SCT CLI mode interaktif](#)
- [Mendapatkan AWS SCT CLI skenario](#)
- [AWS SCT CLI skenario pengeditan](#)
- [AWS SCT CLI mode skrip](#)
- [AWS SCT CLI bahan referensi](#)

Prasyarat untuk menggunakan antarmuka baris perintah AWS SCT

Unduh dan instal versi terbaru Amazon Corretto 11. Untuk informasi selengkapnya, lihat [Unduhan untuk Amazon Corretto 11 di Panduan Pengguna](#) Amazon Corretto 11.

Unduh dan instal versi terbaru AWS SCT. Untuk informasi selengkapnya, lihat [Instalasi AWS Schema Conversion Tool](#).

AWS SCT CLI mode interaktif

Anda dapat menggunakan antarmuka AWS SCT baris perintah dalam mode interaktif. Dalam mode ini, Anda memasukkan perintah ke konsol satu per satu. Anda dapat menggunakan mode interaktif ini untuk mempelajari lebih lanjut tentang CLI perintah atau mengunduh CLI skenario yang paling umum digunakan.

Untuk mengonversi skema basis data sumber Anda AWS SCT, jalankan operasi urutan: buat proyek baru, sambungkan ke basis data sumber dan target, buat aturan pemetaan, dan konversi objek database. Karena alur kerja ini bisa rumit, sebaiknya gunakan skrip dalam mode. AWS SCT CLI Untuk informasi selengkapnya, lihat [Mode skrip](#).

Anda dapat menjalankan AWS SCT CLI perintah dari app folder jalur AWS SCT instalasi Anda. Di Windows, jalur instalasi default adalah `C:\Program Files\AWS Schema Conversion Tool\`. Pastikan folder ini menyertakan `AWSSchemaConversionToolBatch.jar` file.

Untuk masuk ke mode AWS SCT CLI interaktif, gunakan perintah berikut setelah Anda menyelesaikan prasyarat.

```
java -jar AWSSchemaConversionToolBatch.jar -type interactive
```

Sekarang Anda dapat menjalankan AWS SCT CLI perintah. Pastikan Anda mengakhiri perintah Anda / dengan baris baru. Juga, pastikan bahwa Anda menggunakan tanda kutip tunggal (') sebelum dan sesudah nilai parameter perintah.

Note

Jika perintah sebelumnya kembali `Unexpected error`, coba yang berikut ini:

```
java -Djdk.jar.maxSignatureFileSize=20000000 -jar  
AWSSchemaConversionToolBatch.jar
```

Untuk melihat daftar perintah yang tersedia dalam mode AWS SCT CLI interaktif, jalankan perintah berikut.

```
help  
/
```

Untuk melihat informasi tentang AWS SCT CLI perintah, gunakan perintah berikut.

```
help -command: 'command_name'  
/
```

Pada contoh sebelumnya, ganti *command_name* dengan nama sebuah perintah.

Untuk melihat informasi tentang parameter AWS SCT CLI perintah, gunakan perintah berikut.

```
help -command: 'command_name' -parameters: 'parameters_list'  
/
```


Pada contoh sebelumnya, ganti *command_name* dengan nama sebuah perintah. Kemudian, ganti *parameters_list* dengan daftar nama parameter yang dipisahkan dengan koma.

Untuk menjalankan skrip dari file dalam mode AWS SCT CLI interaktif, gunakan perintah berikut.

```
ExecuteFile -file: 'file_path'  
/
```

Pada contoh sebelumnya, ganti *file_path* dengan jalur ke file Anda dengan skrip. Pastikan file Anda memiliki `.scts` ekstensi.

Untuk keluar dari mode AWS SCT CLI interaktif, jalankan `quit` perintah.

Contoh

Contoh berikut menampilkan informasi tentang `Convert` perintah.

```
help -command: 'Convert'  
/
```

Contoh berikut menampilkan informasi tentang dua parameter `Convert` perintah.

```
help -command: 'Convert' -parameters: 'filter, treePath'  
/
```

Mendapatkan AWS SCT CLI skenario

Untuk mendapatkan AWS SCT skenario yang paling umum digunakan, Anda dapat menggunakan `GetCliScenario` perintah. Anda dapat menjalankan perintah ini dalam mode interaktif, lalu mengedit templat yang diunduh. Gunakan file yang diedit dalam mode skrip.

`GetCliScenarioPerintah` menyimpan template yang dipilih atau semua templat yang tersedia ke direktori yang ditentukan. Template berisi set lengkap perintah untuk menjalankan skrip. Pastikan Anda mengedit jalur file, kredensi database, nama objek, dan data lain dalam templat ini. Selain itu, pastikan Anda menghapus perintah yang tidak Anda gunakan dan tambahkan perintah baru ke skrip jika diperlukan.

Untuk menjalankan `GetCliScenario` perintah, selesaikan prasyarat dan masuk ke mode interaktif. AWS SCT CLI Untuk informasi selengkapnya, lihat [Mode interaktif](#).

Selanjutnya, gunakan sintaks berikut untuk menjalankan GetCliScenario perintah dan mendapatkan AWS SCT skenario.

```
GetCliScenario -type: 'template_type' -directory: 'file_path'
/
```

Pada contoh sebelumnya, ganti *template_type* dengan salah satu jenis template dari tabel berikut. Selanjutnya, ganti *file_path* dengan jalur folder tempat Anda ingin mengunduh skrip. Pastikan bahwa AWS SCT dapat mengakses folder ini tanpa meminta hak admin. Juga, pastikan bahwa Anda menggunakan tanda kutip tunggal (' ') sebelum dan sesudah nilai parameter perintah.

Untuk mengunduh semua AWS SCT CLI templat, jalankan perintah sebelumnya tanpa opsi. -type

Tabel berikut mencakup jenis AWS SCT CLI template yang dapat Anda unduh. Untuk setiap template, tabel menyertakan nama file dan deskripsi operasi yang dapat Anda jalankan menggunakan skrip.

Jenis template	Nama file	Deskripsi
BTEQScriptConversion	BTEQScriptConversionTemplate.scts	Mengonversi Teradata Basic Teradata Query (BTEQ),, FastExport FastLoad, dan skrip ke Amazon Redshift. MultiLoad RSQL Untuk informasi selengkapnya, lihat Mengonversi Data Menggunakan ETL .
ConversionApply	ConversionTemplate.scts	Mengonversi skema basis data sumber dan menerapkan kode yang dikonversi ke database target. Secara opsional, menyimpan kode yang dikonversi sebagai SQL skrip, dan menyimpan laporan penilaian. Untuk informasi selengkapnya, lihat Mengkonversi skema .

Jenis template	Nama file	Deskripsi
GenericAppConversion	GenericApplicationConversionTemplate.scts	Mengonversi SQL kode yang disematkan ke dalam aplikasi Anda dengan konverter AWS SCT aplikasi generik. Untuk informasi selengkapnya, lihat SQLkode .
HadoopMigration	HadoopMigrationTemplate.scts	Memigrasi kluster Hadoop lokal Anda ke Amazon. EMR Untuk informasi selengkapnya, lihat Menghubungkan ke database Apache Hadoop dengan AWS Schema Conversion Tool .
HadoopResumeMigration	HadoopResumeMigrationTemplate.scts	Melanjutkan migrasi terputus dari kluster Hadoop lokal Anda ke Amazon. EMR Untuk informasi selengkapnya, lihat Menghubungkan ke database Apache Hadoop dengan AWS Schema Conversion Tool .
Informatika	InformaticaConversionTemplate.scts	Mengonversi SQL kode yang disematkan ke dalam skrip ekstrak, transformasi, dan load () ETL Informatica Anda. Mengonfigurasi koneksi ke basis data sumber dan target Anda dalam ETL skrip Anda, dan menyimpan skrip yang dikonversi setelah konversi. Untuk informasi selengkapnya, lihat Skrip informatika ETL .

Jenis template	Nama file	Deskripsi
LanguageSpecificAppConversion	LanguageSpecificAppConversionTemplate.scts	Mengonversi SQL kode yang disematkan ke dalam aplikasi C #, C ++, Java, dan Pro*C Anda dengan konverter aplikasi. AWS SCT Untuk informasi selengkapnya, lihat Mengonversi aplikasi SQL .
OozieConversion	OozieConversionTemplate.scts	Mengonversi alur kerja Apache Oozie Anda menjadi. AWS Step Functions Untuk informasi selengkapnya, lihat Menghubungkan ke alur kerja Apache Oozie dengan AWS Schema Conversion Tool .
RedshiftAgent	DWHDataMigrationTemplate.scts	Mengonversi skema gudang data sumber dan menerapkan kode yang dikonversi ke database Amazon Redshift target. Kemudian mendaftarkan agen ekstraksi data, membuat dan memulai tugas migrasi data. Untuk informasi selengkapnya, lihat Migrasi dari gudang data .
ReportCreation	ReportCreationTemplate.scts	Membuat laporan migrasi database untuk beberapa skema database sumber. Kemudian menyimpan laporan ini sebagai CSV PDF file. Untuk informasi selengkapnya, lihat Laporan penilaian .

Jenis template	Nama file	Deskripsi
SQLScriptConversion	SQLScriptConversionTemplate.scts	Mengonversi SQL *Plus atau TSQL skrip ke SQL PL/ dan menyimpan skrip yang dikonversi. Juga, menyimpan laporan penilaian.

Setelah Anda mengunduh AWS SCT CLI templat, gunakan editor teks untuk mengonfigurasi skrip agar berjalan di basis data sumber dan target Anda. Selanjutnya, gunakan mode AWS SCT CLI skrip untuk menjalankan skrip Anda. Untuk informasi selengkapnya, lihat [AWS SCT CLI modus skrip](#).

Contoh

Contoh berikut mengunduh semua template ke dalam C:\SCT\Templates folder.

```
GetCliScenario -directory: 'C:\SCT\Templates'  
/
```

Contoh berikut mengunduh template untuk ConversionApply operasi ke dalam C:\SCT\Templates folder.

```
GetCliScenario -type: 'ConversionApply' -directory: 'C:\SCT\Templates'  
/
```

AWS SCT CLISkenario pengeditan

Setelah Anda mengunduh template skenario, konfigurasi mereka untuk mendapatkan skrip yang berfungsi yang dapat berjalan di database Anda.

Untuk semua templat, pastikan Anda memberikan jalur ke driver untuk basis data sumber dan target Anda. Untuk informasi selengkapnya, lihat [Menginstal JDBC driver untuk AWS Schema Conversion Tool](#).

Pastikan Anda menyertakan kredensi database untuk basis data sumber dan target. Selain itu, pastikan Anda menyiapkan aturan pemetaan untuk menjelaskan pasangan sumber-target untuk proyek konversi Anda. Untuk informasi selengkapnya, lihat [Pemetaan tipe data](#).

Selanjutnya, konfigurasi ruang lingkup operasi yang akan dijalankan. Anda dapat menghapus perintah yang tidak Anda gunakan atau menambahkan perintah baru ke skrip.

Misalnya, Anda berencana untuk mengonversi semua skema di database Oracle sumber Anda ke Postgre. SQL Kemudian Anda berencana untuk menyimpan laporan penilaian migrasi database Anda sebagai PDF dan menerapkan kode yang dikonversi ke database target. Dalam hal ini, Anda dapat menggunakan template untuk `ConversionApply` operasi. Gunakan prosedur berikut untuk mengedit AWS SCT CLI template Anda.

Untuk mengedit AWS SCT CLI template untuk **ConversionApply** operasi

1. Buka `ConversionTemplate.scts` yang Anda unduh. Untuk informasi selengkapnya, lihat [Contoh](#).
2. Hapus `CreateFilter`, `Konversi -filter`, `ApplyToTarget -filter`, `SaveTargetSQLSaveTargetSQLbyStatement`, dan `SaveReportCSV` operasi dari skrip Anda.
3. Untuk `oracle_driver_file` dalam `SetGlobalSettings` operasi, masukkan path ke driver Oracle Anda. Kemudian, untuk `postgresql_driver_file`, masukkan path ke driver Postgre Anda. SQL

Jika Anda menggunakan mesin database lain, gunakan nama yang sesuai untuk pengaturan. Untuk daftar lengkap pengaturan global yang dapat Anda atur dalam `SetGlobalSettings` operasi, lihat Matriks pengaturan global di [Bahan referensi](#).

4. (Opsional) Untuk `CreateProject`, masukkan nama proyek Anda dan lokasi untuk file proyek lokal Anda. Jika Anda memilih untuk melanjutkan dengan nilai default, pastikan bahwa AWS SCT dapat membuat file di `C:\temp` folder tanpa meminta hak admin.
5. Untuk `AddSource`, masukkan alamat IP server basis data sumber Anda. Juga, masukkan nama pengguna, kata sandi, dan port untuk terhubung ke server basis data sumber Anda.
6. Untuk `AddTarget`, masukkan alamat IP server basis data target Anda. Juga, masukkan nama pengguna, kata sandi, dan port untuk terhubung ke server basis data target Anda.
7. (Opsional) Untuk `AddServerMapping`, masukkan objek database sumber dan target yang ingin Anda tambahkan ke aturan pemetaan. Anda dapat menggunakan `sourceTreePath` dan `targetTreePath` parameter untuk menentukan jalur ke objek database. Opsional, Anda dapat menggunakan `sourceNamePath` dan `targetNamePath` untuk menentukan nama-nama objek database. Untuk informasi selengkapnya, lihat Perintah pemetaan server di [Bahan referensi](#)

Nilai default `AddServerMapping` operasi memetakan semua skema sumber dengan database target Anda.

8. Simpan file dan kemudian gunakan mode skrip untuk menjalankannya. Untuk informasi selengkapnya, lihat [Mode skrip](#).

AWS SCT CLI modus skrip

Setelah Anda membuat AWS SCT CLI skrip atau mengedit template, Anda dapat menjalankannya dengan RunSCTBatch perintah. Pastikan Anda menyimpan file Anda dengan CLI skrip sebagai `.scts` ekstensi.

Anda dapat menjalankan AWS SCT CLI skrip dari app folder jalur AWS SCT instalasi Anda. Di Windows, jalur instalasi default adalah `C:\Program Files\AWS Schema Conversion Tool\`. Pastikan folder ini menyertakan `RunSCTBatch.sh` file `RunSCTBatch.cmd` atau. Juga, folder ini harus menyertakan `AWSSchemaConversionToolBatch.jar` file.

Atau, Anda dapat menambahkan path ke `RunSCTBatch` file dalam variabel `PATH` lingkungan pada sistem operasi Anda. Setelah Anda memperbarui variabel `PATH` lingkungan, Anda dapat menjalankan AWS SCT CLI skrip dari folder apa pun.

Untuk menjalankan AWS SCT CLI skrip, gunakan perintah berikut di Windows.

```
RunSCTBatch.cmd --pathtoscts "file_path"
```

Pada contoh sebelumnya, ganti *file_path* dengan jalur ke file Anda dengan skrip.

Untuk menjalankan AWS SCT CLI skrip, gunakan perintah berikut di Linux.

```
RunSCTBatch.sh --pathtoscts "file_path"
```

Pada contoh sebelumnya, ganti *file_path* dengan jalur ke file Anda dengan skrip.

Anda dapat memberikan parameter opsional dalam perintah ini, seperti kredensial basis data, tingkat detail dalam output konsol, dan lainnya. Untuk informasi lebih lanjut, unduh referensi antarmuka baris AWS SCT perintah di [Bahan referensi](#).

Contoh

Contoh berikut menjalankan `ConversionTemplate.scts` script dalam `C:\SCT\Templates` folder. Anda dapat menggunakan contoh ini di Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\SCT\Templates\ConversionTemplate.scts"
```

Contoh berikut menjalankan `ConversionTemplate.scts` script dalam `/home/user/SCT/Templates` direktori. Anda dapat menggunakan contoh ini di Linux.

```
RunSCTBatch.sh --pathtoscts "/home/user/SCT/Templates/ConversionTemplate.scts"
```

AWS SCT CLI bahan referensi

Anda dapat menemukan bahan referensi tentang antarmuka baris AWS Schema Conversion Tool perintah (CLI) dalam panduan berikut: [AWS Schema Conversion Tool CLI Referensi](#).

Catatan rilis untuk AWS Schema Conversion Tool

Bagian ini berisi catatan rilis untuk AWS SCT, dimulai dengan versi 1.0.640.

Catatan rilis untuk AWS SCT Build 676

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan di AWS DMS Schema Conversion Tool SCT ()	Ketersediaan dalam AWS DMS Konversi Skema
Oracle	PostgreSQL QLAuro PostgreSQL	Emulasi fungsi bawaan baru untuk fungsi-fungsi berikut: <ul style="list-style-type: none"> <code>SYS.UTL_RAW.BIT_AND(RAW, RAW)</code> <code>XDB.DBMS_XSLPROCESSOR.CLOB2FILE(CLOB)</code> <code>XDB.DBMS_XSLPROCESSOR.READ2CLOB(VARCHAR2)</code> <code>SYS.UTL_RAW.BIT_OR(RAW, RAW)</code> <code>SYS.UTL_RAW.BIT_COMPLEMENT(RAW)</code> 	Tidak	Ya
SQL Server	RDSSQL Amazon	Database Mail not supported Pesan yang dihapus dari PDF laporan	Ya	Ya
Oracle	PostgreSQL QLAuro PostgreSQL	Konversi kendala yang diterapkan untuk tabel yang dipartisi.	Ya	Ya

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan di AWS DMS Schema Conversion Tool SCT ()	Ketersediaan dalam AWS DMS Konversi Skema
Oracle	Saya SQL	Tinjauan penerapan AI-602 dalam konversi tabel	Ya	Ya
SQL Server	PostgreSQL Aurora PostgreSQL	sekarang mendukung MERGE Pernyataan di PostgreSQL 15.x SQL	Ya	Ya
Semua	Semua	JDBC Koneksi yang Diimplementasikan: Properti lanjutan	Ya	Tidak
Semua	Semua	CLI: Kegagalan <code>PrintOLAPTaskStatus</code> perintah tetap	Ya	Tidak
Teradat	Amazon Redshift	Pelakon tipe data gaya Teradata yang diterapkan.	Ya	Tidak
Teradat	Amazon Redshift	Memperbaiki MERGE Konversi yang salah di SQL/BTEQ.	Ya	Tidak
Teradat	Amazon Redshift	Pelakon tipe data gaya Teradata yang diterapkan.	Ya	Tidak
Teradat	Amazon Redshift	Konversi LEAD/LAG fungsi yang diimplementasikan.	Ya	Tidak
Teradat	Amazon Redshift	Kesalahan tetap AI-9996 <code>Transformer error occurred in statement .</code>	Ya	Tidak
Teradat	Amazon Redshift	Kesalahan tetap AI-9996 <code>Transformer error in selectItem .</code>	Ya	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan di AWS DMS Schema Conversion Tool SCT ()	Ketersediaan dalam AWS DMS Konversi Skema
Teradat	Amazon Redshift	Konversi yang diterapkan untuk prosedur tersimpan sebagian: XbidQM.Sp CmprsnDly	Ya	Tidak
Teradat	Amazon Redshift	UNPIVOT Pernyataan terimplementasi dengan alias.	Ya	Tidak
Teradat	Amazon Redshift	Delete Pernyataan yang diimplementasikan dengan beberapa tabel sumber.	Ya	Tidak
Teradat	Amazon Redshift	Perbaiki untuk AI-9996 Transformer error occurred in functionCallExpression .	Ya	Tidak
Teradat	Amazon Redshift	Konversi NORMALIZE klausa yang diterapkan.	Ya	Tidak
Teradat	Amazon Redshift	Memperbaiki konversi yang salah dalam DELETE pernyataan dengan subquery.	Ya	Tidak
Teradat	Amazon Redshift	Kesalahan tetap AI-9996 Transformer error occurred in tableOperatorSource .	Ya	Tidak
Teradat	Amazon Redshift	Kesalahan tetap AI-9996 Transformer error occurred in additiveExpression .	Ya	Tidak
Teradat	Amazon Redshift	Konversi objek DBC sistem yang diterapkan.	Ya	Tidak
Teradat	Amazon Redshift	Solusi pembaruan yang diterapkan dengan predikat gabungan implisit.	Ya	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan di AWS DMS Schema Conversion Tool SCT ()	Ketersediaan dalam AWS DMS Konversi Skema
Netezza	Amazon Redshift	Kesalahan konversi CREATE MATERIALIZED VIEW pernyataan tetap.	Ya	Tidak
dB2luw	PostgreSQL Aurora PostgreSQL	JDBC Koneksi Opsi Diperpanjang: Menambahkan opsi koneksi tambahan.	Ya	Tidak
dB2luw	PostgreSQL Aurora PostgreSQL	Ditambahkan dukungan untuk MERGE Pernyataan di PostgreSQL 15.x SQL	Ya	Tidak
dB2luw	PostgreSQL Aurora PostgreSQL	GLOBAL TEMPORARY TABLE Konversi yang diimplementasikan.	Ya	Tidak
dB2luw	PostgreSQL Aurora PostgreSQL	USER DEFINED TYPES Konversi yang diimplementasikan.	Ya	Tidak
dB2luw	Saya SQL	GLOBAL TEMPORARY TABLE Konversi yang diimplementasikan.	Ya	Tidak
dB2luw	Saya SQL	USER DEFINED TYPES Konversi yang diimplementasikan.	Ya	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan di AWS DMS Schema Conversion Tool SCT ()	Ketersediaan dalam AWS DMS Konversi Skema
dB2luw	Saya SQL	USER DEFINED FUNCTIONS Konversi yang diimplementasikan.	Ya	Tidak
dB2luw	MariaDB	GLOBAL TEMPORARY TABLE Konversi yang diimplementasikan.	Ya	Tidak
dB2luw	MariaDB	USER DEFINED TYPES Konversi yang diimplementasikan.	Ya	Tidak
Sybase	Semua	Menambahkan dukungan untuk otentikasi Kerberos	Ya	Tidak
dB2luw	PostgreSQL QLAuro PostgreSQL	Menambahkan dukungan untuk konversi multi-versi untuk target	Ya	Tidak
SQLAzul Server Microsoft SQL	PostgreSQL QLAuro PostgreSQL	Menambahkan dukungan untuk konversi multi-versi untuk target	Ya	Tidak
dB2luw	PostgreSQL QLAuro PostgreSQL	Menambahkan dukungan untuk MERGE pernyataan di PostgreSQL 15.xSQL.	Ya	Tidak
Teradat	Amazon Redshift	Memperbaiki konversi perubahan fungsi yang tidak didukung.	Ya	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan di AWS DMS Schema Conversion Tool SCT ()	Ketersediaan dalam AWS DMS Konversi Skema
Semua	Amazor Redshif	Ekstraktor Data: Diimplementasikan partisi oleh kolom yang diindeks.	Ya	Tidak

Catatan rilis untuk AWS SCT Build 675

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Cassandra	DynamoD	Memperbaiki bug di mana instalasi Cassandra akan gagal pada Datacenter target.	Tidak
DB2 LUW	Postgre SQL	DYNAMICSQL: PREPARE pernyataan: Selesaikan dan Konversi tanpa DinamisSQL.	Tidak
DB2 LUW	Postgre SQL	Menambahkan dukungan untuk SPECIALREGISTER.	Tidak
DB2 LUW	Postgre SQL	Pembaruan paket ekstensi	Tidak
Hadoop	Amazon EMR	Menambahkan dukungan untuk menghubungkan ke cluster Hadoop melalui protokol rsa-sha2.	Tidak
SQLServer Microsoft	Amazon Redshift	Perbaiki untuk pemaksaan JDBC Driver TLS meskipun tidak dikonfigurasi.	Tidak
Netezza	Amazon Redshift	Menambahkan dukungan untuk konversi tampilan Terwujud.	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Oracle	Amazon Redshift	Menambahkan dukungan untuk kueri rekursif di Amazon Redshift.	Ya
Oracle	Postgre, SQL Aurora PostgreSQL	Perbaiki untuk konversi NUMBER tipe data yang salah.	Ya
Oracle	Amazon Redshift	Migrasi data. Partisi otomatis Oracle. Ditambahkan waktu kedaluwarsa untuk nilai fragmen tabel. Waktu kedaluwarsa adalah 72 jam. Saat kedaluwarsa terjadi, fragmen data dibangun kembali saat tugas migrasi data dibuat.	Tidak
Oracle	Amazon Redshift	SCTData Extractor: Mengubah pendekatan mengunggah data ke Amazon Redshift. Secara default, extractor tidak membuat tabel bertahap. Sebagai gantinya, setelah semua file data berada di bucket Amazon S3, ekstraktor menyalinnya ke tabel target menggunakan satu perintah. COPY	Tidak
Oracle	Amazon Redshift	Menambahkan migrasi tipe RAW data ke VARBYTE kolom.	Tidak
Oracle	Postgre, SQL Aurora PostgreSQL	Konversi multi-versi	Tidak
Oracle	Postgre SQL	Ditambahkan dukungan untuk MERGE Pernyataan di Postgre 15.xSQL.	Ya

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Oracle	Postgre SQL	Menambahkan dukungan untuk fungsi ekspresi reguler baru di Postgre 15.xSQL.	Ya
Oracle	Postgre, SQL Aurora Postgre SQL	UPDATE Pernyataan ON CONFLICT DO dikonversi tanpa alias yang dikecualikan.	Ya
Teradata	Amazon Redshift	Menambahkan dukungan konversi LEAD LAG untuk fungsi.	Tidak
Teradata	Amazon Redshift	Pengecoran tipe data yang disempurnakan dengan indikasi eksplisit format data.	Tidak
Teradata	Amazon Redshift	Peningkatan konversi klausa AT 'TIMEZONE' dalam ekspresi waktu/stempel waktu.	Tidak
Teradata	Amazon Redshift	AI-9996 selama prosedur konversi dengan MERGE pernyataan.	Tidak

Catatan rilis untuk AWS SCT Build 674

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Semua	Semua	Berbagai perbaikan bug dan peningkatan kinerja	Sebagian (hanya untuk pasangan sumber dan target yang didukung)

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
SQL Azure Server Microsoft SQL	Amazon Redshift	Menghapus pesan "AI 18066: Tidak dapat mengonversi nama skema" yang menyesatkan pengguna selama penilaian/konversi skema	Tidak
SQL Azure Server Microsoft SQL	Amazon RDS untuk SQL Saya/ Amazon Aurora Saya SQL	Konversi prosedur yang salah tanpa menetapkan kode pengembalian	Sebagian (Konversi Skema saat ini tidak mendukung Azure SQL sebagai sumber)
SQL Azure Server Microsoft SQL	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	Diperbaiki AI9997 untuk beberapa kasus konversi FOR XML PATH klausa	Sebagian (Konversi Skema saat ini tidak mendukung Azure SQL sebagai sumber)

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
SQL Azure Server Microsoft SQL	Amazon RDS untuk SQL PostgreSQL Amazon Aurora PostgreSQL	Nilai dibulatkan ke skala asli dalam prosedur/fungsi tubuh	Sebagian (Konversi Skema saat ini tidak mendukung Azure SQL sebagai sumber)
SQL Azure Server Microsoft SQL	Amazon RDS untuk SQL PostgreSQL Amazon Aurora PostgreSQL	Berbagai perbaikan konversi EXECUTE pernyataan	Sebagian (Konversi Skema saat ini tidak mendukung Azure SQL sebagai sumber)
SQL Server Microsoft / Azure Synapse	Amazon Redshift	Peningkatan konversi dari pernyataan dan mode berikut: <ul style="list-style-type: none"> EXCEPTION BLOCK AUTOCOMMIT NONATOMIC GROUPING SET CUBE ROLLUP 	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
DB2 LUW	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	Berbagai perbaikan dalam metadata memuat sql-query	Tidak
DB2 LUW	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	AI 9996 tidak diharapkan pada pemacu	Tidak
DB2z/ OS	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	ROWNUMBERfungsi analitik	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
DB2z/ OS	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	Dukungan konstan string heksadesimal	Tidak
DB2z/ OS	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	Berbagai perbaikan dalam metadata memuat sql-query.	Tidak
DB2z/ OS	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	NEXTVALUEFORdukungan referensi urutan	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
DB2z/ OS	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	GETDIAGNOSTICSpernyataan DB2 _ NUMBER _ dukungan ROWS opsi	Tidak
DB2z/ OS	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	GETDIAGNOSTICSBeberapa pernyataan	Tidak
DB2z/ OS	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	Memperbaiki bug dalam konversi FOR pernyataan.	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Oracle	Amazon RDS untuk SQL Saya/ Amazon Aurora Saya SQL	Memperbaiki bug saat node parameter fungsi paket tidak didefinisikan.	Ya
Oracle	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	Memperbaiki bug dalam fungsi paket ekstensi AWS _ ORACLE _EXT. NEXT_ DAY	Ya
Oracle	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	Memperbaiki berbagai bug dengan konversi “(+)” di gabungan luar Oracle	Ya
Oracle		Support Otentikasi Kerberos	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
SAP ASE	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	Memperbaiki bug saat mengonversi lebih dari satu pengenal dalam FROM klausa dalam pernyataan UPDATE	Tidak
SAP ASE	Amazon RDS untuk SQL Postgre/ Amazon Aurora Postgre SQL	Memperbaiki bug dengan komentar multi-baris dan konversi pernyataan	Tidak
SAP ASE		Menambahkan dukungan untuk PASSWORD parameter ENCRYPT _ saat menghubungkan	Tidak
Teradata	Amazon Redshift	Peningkatan konversi VOLATILE tabel dengan nama skema tertentu	Tidak
Teradata	Amazon Redshift	Konversi yang salah WHERE CLAUSE dalam kompleks CTE	Tidak
Teradata	Amazon Redshift	Menambahkan dukungan untuk INTERVAL tipe data saat memigrasi data menggunakan SCT Agen Ekstraksi Data.	Tidak

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki	Ketersediaan dalam AWS DMS Konversi Skema
Skrip Teradata BTEQ	Skrip Amazon Redshift RSQL	Parameter konversi keluar yang salah dalam prosedur yang dijalankan oleh BTEQ	Tidak

Catatan rilis untuk AWS SCT Build 673

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Perbaiki bug umum dan peningkatan kinerja
SQLAzure/ Server Microsoft SQL	Aurora Postgre/ Amazon Postgre SQL RDS SQL	Memperbaiki konversi panggilan fungsi yang salah
SQLAzure/ Server Microsoft SQL	Aurora Postgre/ Amazon Postgre SQL RDS SQL	Konversi FOR XML klausa yang diterapkan
SQLAzure/ Server Microsoft SQL	Aurora Postgre/ Amazon Postgre SQL RDS SQL	Konversi FOR XML klausa dengan alias yang salah.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLAzure/ Server Microsoft SQL	Aurora Postgre/ Amazon Postgre SQL RDS SQL	Memperbaiki bug saat AWS SCT tidak mengonversi EXECUTE pernyataan yang menjalankan string karakter dengan parameter prosedur.
SQLAzure/ Server Microsoft SQL	Aurora Postgre/ Amazon Postgre SQL RDS SQL	Peningkatan konversi UPDATE pernyataan dengan gabungan batin.
Sinaps Azure	Amazon Redshift	Memperbaiki konversi fungsi OBJECT_ID bawaan yang salah.
IBMDB2 untuk z/OS	Aurora Postgre/ Amazon Postgre SQL RDS SQL	Menerapkan konversi pernyataan dan objek berikut: <ul style="list-style-type: none"> • DECLARE TEMPORARY TABLE statement • DROP TABLE statement • PK dan UNIQUE kendala pada tabel yang dipartisi • Fungsi TIMESTAMPDIFF • Fungsi TO_DATE • Fungsi EBCDIC_STR • Fungsi VARCHAR_FORMAT
IBMDB2 untuk z/OS	Aurora Postgre/ Amazon Postgre SQL RDS SQL	Memperbaiki bug saat indeks berbasis fungsi melewati fungsi setelah konversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDB2 untuk z/OS	Aurora PostgreSQL Amazon PostgreSQL RDS SQL	Memperbaiki bug di mana REPEAT pernyataan ditutup dengan AI 9996 setelah konversi
IBMDB2 untuk z/OS	Aurora PostgreSQL Amazon PostgreSQL RDS SQL	Memperbaiki bug di mana FINAL TABLE klausa ditutup dengan 9996.
IBMDB2 untuk z/OS	Aurora PostgreSQL Amazon PostgreSQL RDS SQL	LOADER Kunci partisi dalam batasan referensi. AWS SCT sekarang dapat mengonversi kunci primer dan kendala unik dalam tabel yang dipartisi sebagai indeks sekunder.
IBMDB2 untuk z/OS	Aurora PostgreSQL Amazon PostgreSQL RDS SQL	PostgreSQL. VARCHAR_ dukungan FORMAT fungsi
IBMDB2 untuk z/OS	Aurora PostgreSQL Amazon PostgreSQL RDS SQL	Menerapkan perubahan pemeriksaan CreateTransformationRule dan ModifyTransformationRule SCT CLI perintah.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Greenplum	Amazon Redshift	Memperbaiki bug dengan panggilan prosedur tersimpan yang salah setelah konversi
Hadoop	Amazon EMR	Menambahkan dukungan untuk menghubungkan ke cluster Hadoop menggunakan protokol rsa-sha2.
Hadoop	Amazon EMR	Menambahkan dukungan untuk Amazon EMR dengan metastore Non-glue Hive,
Oracle	Amazon Redshift	Memperbaiki bug dengan konversi kueri rekursif yang salah di mana PRIOR kolom tidak ada dalam SELECT daftar.
Oracle	Aurora Postgre/ Amazon Postgre SQL RDS SQL	Diimplementasikan mengembalikan elemen array asosiatif
Oracle	Aurora Postgre/ Amazon Postgre SQL RDS SQL	Memperbaiki AI 9996 yang tidak terduga UNPIVOT dengan tanda kurung
Oracle	Aurora Postgre/ Amazon Postgre SQL RDS SQL	Memperbaiki AI 9996 yang tidak terduga dengan UNPIVOT UNION ALL

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre/ Amazon Postgre SQL RDS SQL	perbaikan untuk konversi tipe Number data
Oracle	Ekstraktor Data Pergeseran Merah Amazon	Support untuk partisi otomatis untuk tabel Oracle. Optimalisasi untuk membuat tugas migrasi.
Teradata	Amazon Redshift	Menerapkan konversi EXCEPTION BLOCK pernyataan
Teradata	Amazon Redshift	Support untuk konversi ALL, ANY, dan SOME predikat ke Amazon Redshift.
Teradata	Amazon Redshift	Menambahkan dukungan asli untuk QUALIFY predikat.
Teradata	Amazon Redshift	Peningkatan konversi berikut ini: <ul style="list-style-type: none"> • Kueri rekursif • GROUPING SET • CUBE • ROLLUP • UPDATE pernyataan dengan gabungan implisit
OLAP sumber	Ekstraktor Data Pergeseran Merah Amazon	CLI Perintah yang diterapkan untuk Stop/Resume untuk tugas Amazon Redshift Data Extractor.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
OLAP sumber	Ekstraktor Data Pergeseran Merah Amazon	Menambahkan kemampuan untuk memilih kolom tabel yang perlu dimigrasi selama konfigurasi tugas migrasi.

Catatan rilis untuk AWS SCT Build 672

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Amazon RDS para PostgreSQL	Menerapkan dukungan PostgreSQL mayor versi 15 sebagai target migrasi.
Semua	Amazon Redshift	Menambahkan <code>PrintTaskStatus</code> perintah baru di antarmuka baris AWS SCT perintah (CLI) untuk menampilkan status tugas migrasi data.
Semua	Amazon Redshift	Meningkatkan aliran konfigurasi untuk agen ekstraksi data.
Semua	Amazon Redshift	Mengatasi kesalahan di mana agen ekstraksi data tidak menampilkan informasi tentang subtugas.
Apache Oozie	AWS Step Functions	Menambahkan opsi untuk menyimpan definisi mesin negara sebagai skrip dalam kode yang dikonversi.
Database Azure SQL Server Microsoft	Aurora PostgreSQL PostgreSQL	Menerapkan konversi <code>COALESCE</code> , <code>DATEADD</code> , <code>GETDATE</code> , dan <code>SUM</code> fungsi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi UPDATE pernyataan dengan JOIN dan OUTPUT klausa.
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Menyelesaikan kesalahan yang terjadi selama konversi SELECT TOP 1 WITH TIES pernyataan.
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Menyelesaikan beberapa masalah yang terjadi selama konversi FOR XML klausa dalam fungsi bawaan.
Greenplum	Amazon Redshift	Menerapkan konversi GET DIAGNOSTICS dan RAISE pernyataan dengan menggunakan blok Amazon Redshift EXCEPTION asli.
Greenplum	Amazon Redshift	Meningkatkan konversi prosedur tersimpan dengan menambahkan dukungan EXCEPTION blok dalam kode yang dikonversi.
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan di mana TO_CHAR fungsi dengan templat format waktu tidak dikonversi dengan benar.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menerapkan konversi ekspresi tabel bersarang.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menerapkan konversi GOTO, MERGE, REPEAT, dan SIGNAL pernyataan.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menerapkan konversi FETCH pernyataan dengan BEFORE dan kata kunci AFTER orientasi.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menerapkan konversi referensi FINAL TABLE dan OLD TABLE tabel.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	<p>Menerapkan konversi fungsi-fungsi berikut.</p> <ul style="list-style-type: none"> • ADD_MONTHS • DAY dengan parameter tipe data karakter • DAYOFWEEK • DAYS • DECODE • HOUR • LAST_DAY • LOCATE_IN_STRING • MICROSECOND • MINUTE • MONTH • ROUND • TIME • TIMESTAMP • TIMESTAMP_FORMAT • TRANSLATE • UNICODE_STR • XMLCAST • XMLELEMENT • XMLQUERY • XMLSERIALIZE • YEAR
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi alias subquery dalam JOIN klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi COALESCE fungsi.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi EXPLICIT indeks.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi nama kolom dalam ekspresi majemuk untuk menyelesaikan masalah di mana item tindakan 9997 tiba-tiba muncul selama konversi.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi kunci primer dan kendala unik.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Memperbaiki konversi XMLTABLE pernyataan dalam INSERT pernyataan untuk menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana item tindakan 9996 muncul secara tak terduga selama konversi fungsi dengan argumen. SUBSTR
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi register khusus. CURRENT_TIMESTAMP
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah saat item tindakan 9996 muncul secara tak terduga selama konversi MERGE pernyataan, pernyataan yang tidak didukung, dan fungsi bawaan yang tidak didukung.
SQLServer Microsoft	Semua	Menambahkan dukungan Microsoft SQL Server versi 2022 sebagai sumber.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi SELECT pernyataan yang menggunakan operator penggabungan string. AWS SCT menggunakan STRING_AGG fungsi dalam kode yang dikonversi.
SQLServer Microsoft	Babelfish untuk Aurora Postgre SQL	Menerapkan dukungan versi baru 3.1.0 dari file konfigurasi fitur Babelfish. File ini mendefinisikan SQL fitur yang didukung dan tidak didukung oleh versi Babelfish tertentu.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Netezza	Amazon Redshift	Menyelesaikan masalah di mana agen ekstraksi data tidak memulai migrasi data dari CDC titik yang ditentukan.
Oracle	Semua	Memperbarui laporan penilaian untuk database Oracle versi 19 sebagai sumber.
Oracle	Aurora PostgreSQL PostgreSQL	Menerapkan konversi DBMS_OUTPUT paket dengan menambahkan fungsi baru ke paket AWS SCT ekstensi.
Oracle	Aurora PostgreSQL PostgreSQL	Menerapkan konversi fungsi dan prosedur yang menggunakan array asosiatif sebagai argumen atau parameter.
Oracle	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi DISTINCT klausa dalam SELECT pernyataan.
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan konversi tabel di mana kendala kunci utama memiliki nama yang sama dengan tabel.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi RAISE_APPLICATION_ERROR prosedur dengan parameter ketiga.
Oracle	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana aturan migrasi tidak secara otomatis mengubah tipe NUMERIC data ke INTEGER tempat yang berlaku.
Oracle DW	Amazon Redshift	Menerapkan dukungan CONNECT BY klausa Amazon Redshift asli dalam kode yang dikonversi.
Oracle DW	Amazon Redshift	Meningkatkan migrasi data dengan secara otomatis menambahkan subtugas untuk setiap tabel atau partisi dalam lingkup migrasi. Pendekatan ini mencegah kehilangan data untuk data yang dimasukkan setelah partisi.
Teradata	Amazon Redshift	Menerapkan konversi pandangan rekursif.
Teradata	Amazon Redshift	Meningkatkan konversi prosedur tersimpan yang menggunakan BTET dan mode ANSI transaksi dengan menambahkan dukungan mode AUTOCOMMIT transaksi Amazon Redshift asli.
Teradata	Amazon Redshift	Memperbaiki konversi prosedur tersimpan yang menggunakan semantik TERADATA transaksi dengan menambahkan NONATOMIC kata kunci dalam kode yang dikonversi.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana kode yang dikonversi menyertakan ID kunci AWS akses dan kunci akses rahasia.

Catatan rilis untuk AWS SCT Build 671

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Memperbaiki kesalahan di mana AWS SCT tidak memiliki izin untuk menyimpan file proyek di Windows.
Semua	Semua	<p>Memperbarui template antarmuka baris AWS SCT perintah (CLI) berikut.</p> <ul style="list-style-type: none"> • BTEQScriptConversion • ConversionApply • HadoopMigration • HadoopResumeMigration • Informatika <p>Untuk informasi selengkapnya tentang AWS SCT CLI template, lihat Mendapatkan CLI skenario.</p>
Semua	Amazon Redshift	Memperbaiki kesalahan saat AWS SCT tidak membuat paket ekstensi di antarmuka baris perintah (CLI).
Semua	Amazon Redshift	Menyelesaikan masalah di mana agen ekstraksi AWS SCT data tidak menggunakan AWS Snowball konfigurasi di antarmuka baris perintah (CLI).
Apache Oozie	AWS Step Functions	Menerapkan dukungan untuk migrasi dari Apache Oozie ke AWS Step Functions dalam mode antarmuka baris perintah (CLI). Setelah memigrasikan beban kerja Hadoop Anda ke EMR Amazon, Anda sekarang dapat memigrasikan sistem penjadwalan alur kerja ke AWS Cloud. Untuk informasi selengkapnya, lihat Mengonversi alur kerja Oozie .
Database Azure SQL	Aurora PostgreSQL	Memperbaiki kesalahan resolver yang terjadi untuk tabel dan alias.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	Postgre SQL	
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Menerapkan konversi INDEX ON klausa.
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi objek berikut untuk menghindari item tindakan yang tidak terduga. <ul style="list-style-type: none"> • Pernyataan Batch • Daftar ekspresi • Alias tabel • Tabel sementara • Pemicu • Variabel pengguna
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Mengatasi kesalahan penguraian yang terjadi untuk prosedur.
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan di mana AWS SCT digunakan nama tabel sementara yang salah dalam kode yang dikonversi untuk OBJECT_ID fungsi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Masalah yang diselesaikan di mana item tindakan 9996 tiba-tiba muncul selama konversi elemen kode berikut. <ul style="list-style-type: none"> • Fungsi CONVERT • Fungsi DATEADD • DELETEpernyataan di dalam fungsi inline • Pernyataan IF • INSERTatau UPDATE tindakan pada kolom • Pernyataan RETURN • UPDATEpernyataan dengan kueri atau fungsi yang kompleks
BigQuery	Amazon Redshift	Menambahkan dukungan untuk BigQuery sebagai sumber untuk proses penilaian multiserver. Untuk informasi selengkapnya, lihat Laporan penilaian multiserver .
Hadoop	Amazon EMR	Memperbarui versi JDBC driver Apache Hive yang didukung yang Anda gunakan untuk terhubung ke database sumber Anda. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool .
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Meningkatkan pemuat metadata sumber untuk memastikan bahwa AWS SCT memuat objek database sumber seperti kunci utama, indeks implisit, dan sebagainya.
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan resolver yang terjadi untuk kolom dalam kursor implisit.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan kemampuan untuk menjaga pemformatan nama kolom, ekspresi, dan klausa dalam DML pernyataan dalam kode yang dikonversi.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan konversi kunci asing skema silang.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan konversi LENGTH dan VARCHAR fungsi.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan konversi LABEL ON dan DECLARE CONDITION pernyataan.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan konversi SELECT pernyataan dengan OPTIMIZE FOR klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi CREATE TABLE pernyataan dengan menambahkan nilai default untuk semua tipe data yang didukung.
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi INCREMENT BY atribut.
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi tabel dipartisi dengan menambahkan kemampuan untuk mengecualikan partisi tabel dari lingkup konversi.
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi definisi kunci primer dengan INCLUDE kolom.
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi SUBSTRING fungsi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi SET dan DECLARE HANDLER FOR pernyataan.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi tipe data variabel.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan konversi XMLTABLE fungsi.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Meningkatkan aliran migrasi dengan menerapkan urutan berikut menerapkan objek dikonversi ke database target: tabel, partisi, indeks, kendala, kunci asing, dan pemicu.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi komentar di kode sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah saat item tindakan 9997 muncul secara tak terduga selama konversi alias dalam klausa. FROM
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah saat item tindakan 9997 muncul secara tak terduga selama konversi alias cursor.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan di mana kode yang dikonversi mengembalikan hasil yang berbeda untuk SELECT pernyataan dengan ORDER BY klausa. Karena SQL Server dan Postgre SQL memperlakukan NULL nilai secara berbeda, kode yang dikonversi sekarang menyertakan NULLS FIRST atau NULLS LAST klausa yang memastikan bahwa kode yang dikonversi mengembalikan hasil dalam urutan yang sama dengan kode sumber Anda.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana tipe data dalam fungsi tabel salah dikonversi.
Saya SQL	Amazon RDS untuk Saya SQL	Menyelesaikan masalah di mana tanda kutip tunggal (' ') tiba-tiba muncul di sekitar nama objek database dalam kode yang dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Menambahkan tampilan baru ke paket ekstensi untuk meniru tampilan sistem Oracle yang Anda gunakan untuk menampilkan informasi tentang partisi dan subpartisi.
Oracle	Aurora Postgre SQL Postgre SQL	Diperbarui dua fungsi dalam paket ekstensi untuk menambahkan nama skema sebagai argumen dalam kode yang dikonversi.
Oracle	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan yang AWS SCT tidak menggunakan parameter yang benar untuk konversi aplikasi C++ setelah menyegarkan kode aplikasi di antarmuka pengguna.
Oracle	Aurora Postgre SQL Postgre SQL	Memperbaiki konversi CREATE TYPE pernyataan untuk menghindari pengecualian yang tidak terduga.
Oracle	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi tabel bersarang.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Menyelesaikan kesalahan penguraian yang terjadi untuk objek paket.
Oracle	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana nama objek dipangkas AWS SCT secara tak terduga dalam kode yang dikonversi ketika panjang nama melebihi 60 karakter.
Oracle	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana pemacu tingkat baris untuk tabel yang dipartisi salah dikonversi.
Oracle DW	Amazon Redshift	Menerapkan dukungan partisi tabel otomatis untuk migrasi data. Untuk mempercepat migrasi data, secara otomatis AWS SCT dapat partisi tabel besar atau partisi berdasarkan nilai-nilai dalam ROWID pseudocolumn. Untuk informasi selengkapnya, lihat Menggunakan partisi asli .
Teradata	Amazon Redshift	Menerapkan dukungan MERGE perintah asli dalam kode Amazon Redshift yang dikonversi. Untuk informasi selengkapnya tentang MERGE perintah di Amazon Redshift, lihat MERGE di Panduan Pengembang Database Amazon Redshift.
Teradata	Amazon Redshift	Memperbaiki konversi DELETE dan UPDATE pernyataan yang tidak menggunakan nama tabel eksplisit.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana IN dan NOT IN pernyataan salah dikonversi.

Catatan rilis untuk AWS SCT Build 670

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	<p>Masalah yang diselesaikan di mana item tindakan 9996 tiba-tiba muncul selama konversi elemen kode berikut.</p> <ul style="list-style-type: none"> • CREATE INDEX pernyataan di dalam INCLUDE pernyataan • Pernyataan DECLARE • Pernyataan DECLARE . . . TABLE • DECLARE dengan nilai default di dalam LOOP pernyataan • Pernyataan DELETE • DROP CONSTRAINT pernyataan di dalam ALTER TABLE pernyataan • EXECUTE AS CALLER dan REVERT • Pernyataan IIF • Daftar ekspresi • Fungsi MONTH() • Pernyataan UPDATE • Fungsi YEAR()
Analisis Sinaps Azure	Amazon Redshift	Menambahkan dukungan untuk Azure Synapse Analytics sebagai sumber untuk proses penilaian multiserver. Untuk informasi selengkapnya, lihat Laporan penilaian multiserver .
Hadoop	Amazon EMR	Menerapkan dukungan untuk migrasi cluster Hadoop ke Amazon EMR dalam mode antarmuka baris perintah ()CLI. Untuk informasi selengkapnya, lihat Migrasi kerangka kerja data besar .
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan resolver yang terjadi untuk tabel sumber dan kolom.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menerapkan konversi CASE ekspresi.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menerapkan konversi CURRENT_DATE referensi ke register khusus. Referensi ke register khusus di Db2 untuk z/OS adalah referensi ke nilai yang disediakan oleh server saat ini.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menerapkan konversi DATE dan POSSTR fungsi.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi konstanta datetime.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi nilai default untuk kolom tipe data berikut: DATE,, TIMESTAMPTIME , danTIMESTAMP WITH TIME ZONE.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana item tindakan 9996 muncul secara tak terduga selama konversi pernyataan. <code>SELECT INTO</code>
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi <code>DATEDIFF</code> fungsi.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan di mana <code>ISNULL</code> fungsi dikonversi ke <code>NULLIF</code> . Akibatnya, kode yang dikonversi menghasilkan hasil yang berbeda dibandingkan dengan kode sumber. Sekarang, AWS SCT mengkonversi <code>ISNULL</code> fungsi ke <code>COALESCE</code> .
Netezza	Amazon Redshift	Agen ekstraksi data yang ditingkatkan untuk menyelesaikan masalah di mana status gagal ditetapkan untuk tugas yang berhasil diselesaikan.
Netezza	Amazon Redshift	Menambahkan kemampuan untuk mengubah titik akhir dalam subtugas setelah memulai migrasi data dengan agen ekstraksi data.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	Aurora Saya SQL	Ditambahkan kemampuan untuk terhubung ke database menggunakan protokol IPv6 alamat.
Saya SQL Oracle	Aurora Postgre SQL	
Postgre SQL	Saya SQL Postgre SQL	
Oracle	Amazon RDS untuk Oracle	Konversi DBMS_JOB paket yang diimplementasikan yang menjadwalkan dan mengelola pekerjaan dalam antrian pekerjaan.
Oracle	Aurora Postgre SQL Postgre SQL	Menambahkan fungsi baru ke paket ekstensi untuk meningkatkan konversi tabel bersarang global. Fungsi-fungsi baru ini meniru DELETE, EXTEND, dan TRIM berfungsi dalam kode Oracle sumber Anda.
Oracle	Aurora Postgre SQL Postgre SQL	Ditambahkan kemampuan untuk menentukan lingkup konversi untuk SQL kode yang tertanam dalam aplikasi Java. Anda sekarang dapat mengecualikan himpunan bagian dari proyek aplikasi sumber dari lingkup konversi. Untuk informasi selengkapnya, lihat Mengonversi SQL kode aplikasi Java Anda di AWS SCT .
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi operator penggabungan () di dalam indeks fungsional.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi IN kondisi di mana kode sumber Anda tidak menyertakan tanda kurung untuk satu ekspresi.
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi MERGE pernyataan ke INSERT ON CONFLICT dalam PostgreSQL.
Oracle	Aurora Postgre SQL Postgre SQL	Mengatasi kesalahan penguraian yang terjadi untuk paket prosedur.
Oracle	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah saat item tindakan 5072 muncul secara tak terduga selama konversi paket.
Oracle DW	Amazon Redshift	Memperbaiki kesalahan yang AWS SCT tidak menerapkan paket ekstensi saat menerapkan kode yang dikonversi ke database target.
Oracle DW	Amazon Redshift	Memperbaiki kesalahan yang AWS SCT tidak menerapkan beberapa file paket ekstensi saat menggunakan wizard paket ekstensi.
Oracle DW	Amazon Redshift	Menyelesaikan masalah yang AWS SCT tidak dapat memproses migrasi data AWS Snowball dengan lebih dari 500 tugas yang berjalan secara paralel.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle DW	Amazon Redshift	Menyelesaikan masalah di mana fungsi yang ditentukan pengguna dengan tipe yang ditentukan pengguna salah dikonversi.

Catatan rilis untuk AWS SCT Build 669

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Meningkatkan proses penilaian multiserver, yang membantu menentukan platform basis data target yang optimal untuk basis data sumber Anda. Sekarang, AWS SCT abaikan AWS Secrets Manager kuncinya jika Anda memberikan kredensi database dalam file input comma separated values (.CSV). Untuk informasi selengkapnya, lihat Laporan penilaian multiserver .
Semua	Semua	Menyelesaikan masalah di mana laporan penilaian multiserver menyertakan alamat IP database sumber Anda saat menggunakan rahasia dari AWS Secrets Manager untuk terhubung ke database.
Semua	Amazon Redshift	Menerapkan konfigurasi otomatis pengaturan mesin virtual Java (JVM) tergantung pada sistem operasi dan tersediaRAM. AWS SCT menggunakan ini JVM untuk menjalankan pekerjaan agen ekstraksi data.
Semua	Amazon Redshift	Menyelesaikan masalah di mana agen ekstraksi data tidak dimulai di Ubuntu.
Semua	Amazon Redshift	Menyelesaikan masalah di mana tugas ekstraksi data tidak dimulai setelah menjalankan StartAgent.bat file di Windows.
Database Azure SQL Server Microsoft	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana nama kolom salah dikonversi dengan opsi Hasilkan nama unik untuk indeks diaktifkan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Greenplum	Amazon Redshift	Menerapkan konversi fungsi yang kembali VOID ke prosedur.
Greenplum	Amazon Redshift	Menyelesaikan masalah saat migrasi data gagal saat database sumber tidak menyertakan nilai numerik (NaN) dalam kolom numerik. AWS SCT agen ekstraksi data sekarang mengganti nilai NaN dengan NULL.
IBMDb2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menambahkan pengaturan konversi baru untuk menentukan DATE FORMAT dan TIME FORMAT opsi selama konversi fungsi CHAR bawaan.
IBMDb2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menambahkan item tindakan 8534 untuk konversi kursor yang telah ditentukan yang dideklarasikan dengan klausa. WITHOUT RETURN Jika kursor Anda tidak mengembalikan kumpulan hasil, AWS SCT berikan NULL nilai ke nama kursor Anda dalam kode yang dikonversi dan memunculkan item tindakan.
IBMDb2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Mengedit CURRENT CLIENT_APPLNAME properti yang mengidentifikasi AWS SCT selama koneksi ke database sumber.
IBMDb2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menerapkan pengaturan konversi baru untuk menentukan DATE FORMAT dan TIME FORMAT opsi selama konversi fungsi CHAR bawaan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Konversi yang diimplementasikan dari LEAVE pernyataan dalam pernyataan BEGIN . . . END blok.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menerapkan konversiXMLPARSE,XMLTABLE, dan XMLNAMESPACES fungsi.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Peningkatan konversi fungsi CHAR bawaan.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Peningkatan konversi kursor.
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi pernyataan loop. FOR

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Peningkatan konversi dari jenis tabel penggunaan dalam SELECT pernyataan.
SQLServer Microsoft	Babelfish untuk Aurora Postgre SQL	Menerapkan dukungan versi baru 2.2.0 dari file konfigurasi fitur Babelfish. File ini mendefinisikan SQL fitur yang didukung dan tidak didukung oleh versi Babelfish tertentu.
Netezza	Amazon Redshift	Agan ekstraksi data yang ditingkatkan untuk menyelesaikan masalah di mana satu baris tidak dihapus dari tabel target selama replikasi data yang sedang berlangsung.
Oracle	Amazon RDS untuk Oracle	Peningkatan konversi fitur Oracle Database Enterprise Edition.
Oracle	Aurora Postgre SQL Postgre SQL	Konversi GROUPING_ID fungsi yang diimplementasikan.
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi SQL kode dalam aplikasi C # dengan menambahkan dukungan pemetaan tipe data kustom dalam mode antarmuka baris perintah (CLI).

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi tabel bersarang untuk menghindari item tindakan yang tidak terduga 9996.
Oracle	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana panggilan konstruktor objek salah dikonversi.
Oracle DW	Amazon Redshift	Menerapkan dukungan partisi tabel yang ada untuk migrasi data. Untuk mempercepat migrasi data, AWS SCT buat subtask untuk setiap partisi tabel sumber yang tidak kosong. Untuk informasi selengkapnya, lihat Menggunakan partisi asli .
Teradata	Amazon Redshift	Peningkatan konversi CAST fungsi dengan <code>TIME WITH TIME ZONE AS TIMESTAMP</code> , <code>TIME WITH TIME ZONE AS CHAR</code> , dan <code>TIMESTAMP AS TIME WITH TIME ZONE</code> argumen.
Teradata	Amazon Redshift	Peningkatan konversi CAST fungsi dengan <code>FORMAT</code> opsi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana <code>CEIL</code> fungsi tidak dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana <code>MERGE</code> pernyataan dengan <code>DELETE</code> klausa salah dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana <code>TO_CHAR</code> fungsi dengan argumen tanggal dan format salah dikonversi.

Catatan rilis untuk AWS SCT Build 668

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Amazon Redshift	Menyelesaikan masalah di mana operator perkalian dalam aturan migrasi tidak berfungsi dengan benar. Operator ini memungkinkan untuk mengubah panjang <code>char</code> , <code>varchar</code> , dan tipe <code>string</code> data. Untuk informasi selengkapnya, lihat Membuat aturan migrasi .
Analisis Sinaps Azure	Amazon Redshift	Menerapkan dukungan CONVERT fungsi dengan VARCHAR argumen.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi SELECT pernyataan dengan NOLOCK klausa.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi UPDATE pernyataan dengan alias atau dengan SET dan FROM klausa.
Greenplum	Amazon Redshift	Menerapkan partisi virtual otomatis untuk migrasi data. AWS SCT menggunakan kolom <code>GP_SEGMENT_ID</code> sistem untuk membuat partisi.
Greenplum	Amazon Redshift	Menerapkan dukungan RETURN QUERY dan RETURN SETOF klausa.
Greenplum	Amazon Redshift	Menerapkan dukungan SUBSTRING fungsi dengan tiga parameter.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Peningkatan konversi SUBSTR fungsi dengan LOCATE parameter.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Ditambahkan pilihan untuk menggunakan array REFCURSOR variabel untuk mengembalikan set hasil dinamis. Saat Anda memilih opsi ini di pengaturan konversi, AWS SCT tambahkan OUT parameter tambahan dalam kode yang dikonversi.
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Dukungan yang diimplementasikan dari pernyataan FOR loop.
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menerapkan dukungan XMLPARSE fungsi. Menambahkan item tindakan 8541 untuk striping spasi putih dalam XMLPARSE fungsi.
IBMDb2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Meningkatkan konversi beberapa penanganan pengecualian dalam satu BEGIN . . . END blok.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Peningkatan konversi INSERT dan DELETE pemicu.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Peningkatan konversi panggilan prosedur bersarang.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Peningkatan konversi jenis tabel.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana NOT operasi logis bitwise salah dikonversi untuk nilai integer.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana array lokal tidak diinisialisasi di Postgre SQL versi 8.0.2 dan yang lebih rendah.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana MERGE pernyataan dengan WHEN NOT MATCHED BY SOURCE klausa salah dikonversi.
Saya SQL	Aurora Saya SQL	Menyelesaikan masalah ketika AWS SCT salah menentukan izin pengguna yang diberikan oleh <code>rds_superuser_role</code> peran.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Netezza	Amazon Redshift	Meningkatkan pemuat metadata sumber untuk memastikan bahwa memuat objek database AWS SCT dengan benar dengan nama dalam huruf kecil.
Oracle	Aurora PostgreSQL PostgreSQL	Menambahkan fungsi baru ke paket ekstensi untuk meningkatkan konversi tabel bersarang lokal. Fungsi-fungsi baru ini meniru PRIORNEXT, LIMIT, FIRST, LAST, EXISTS, EXTEND, TRIM, DELETE, dan SET fungsi dalam kode Oracle sumber Anda. Untuk informasi selengkapnya, lihat Paket ekstensi .
Oracle	Aurora PostgreSQL PostgreSQL	Ditambahkan kemampuan untuk menentukan lingkup konversi untuk aplikasi C#. Pengguna sekarang dapat mengecualikan himpunan bagian dari proyek aplikasi sumber dari lingkup konversi.
Oracle	Aurora PostgreSQL PostgreSQL	Menerapkan dukungan COUNT metode dalam koleksi.
Oracle	Aurora PostgreSQL PostgreSQL	Menerapkan dukungan variabel dan konstruktor dalam tabel bersarang.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Menerapkan dukungan <code>RATIO_TO_REPORT</code> dan <code>STANDARD_HASH</code> fungsi.
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi objek besar (LOBs) sebagai bagian dari paket AWS SCT ekstensi.
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi koleksi lokal.
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi JOIN pernyataan dengan USING klausa di mana nama kolom tidak menyertakan nama tabel.
Oracle	Aurora Postgre SQL Postgre SQL	Konversi <code>EMPTY_BLOB</code> dan <code>EMPTY_CLOB</code> fungsi yang diterapkan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Konversi yang diimplementasikan dari variabel pengikat posisi dalam aplikasi C #.
SAP ASE	Aurora Postgre SQL Postgre SQL	Konversi yang diimplementasikan dari pemacu multi-peristiwa.
SAP ASE	Aurora Postgre SQL Postgre SQL	Konversi yang diimplementasikan dari pemacu rekursif.
SAP ASE	Aurora Postgre SQL Postgre SQL	Peningkatan konversi pemacu dengan variabel @@rowcount global.
SAP ASE	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana fungsi agregat dalam SET klausa UPDATE pernyataan salah dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SAP ASE	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana item tindakan 42702 tiba-tiba muncul selama konversi pernyataan. UPDATE
SAP ASE	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana CONVERT fungsi dengan CHAR argumen salah dikonversi.
Kepingan salju	Amazon Redshift	Menambahkan dukungan Snowflake sebagai sumber migrasi data dengan agen ekstraksi AWS SCT data. Untuk informasi selengkapnya, lihat Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool .
Teradata	Amazon Redshift	Peningkatan konversi CAST fungsi dengan TIMESTAMP AS TIME WITH TIMEZONE argumen.

Catatan rilis untuk AWS SCT Build 667

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menerapkan dukungan Informatica ekstrak, transformasi, dan load (ETL) skrip dalam mode antarmuka baris perintah (CLI). AWS SCT secara otomatis mengalihkan ETL skrip Informatica Anda ke database target baru. Juga, AWS SCT mengkonversi nama objek dan SQL kode yang tertanam dalam objek Informatica Anda. Untuk informasi selengkapnya, lihat Skrip informatika ETL .

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Amazon Redshift	Meningkatkan versi driver minimum yang didukung untuk Amazon Redshift menjadi 2.1.0.9. Untuk informasi selengkapnya, lihat Menginstal JDBC driver untuk AWS Schema Conversion Tool .
Analisis Sinaps Azure	Amazon Redshift	Menambahkan fungsi baru ke paket ekstensi untuk meningkatkan konversi CONVERT fungsi dengan tiga argumen tanggal dan waktu.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi DATEDIFF fungsi.
Analisis Sinaps Azure	Amazon Redshift	Memperbarui versi paket ekstensi. Pastikan Anda menerapkan versi terbaru dari paket ekstensi dalam AWS SCT proyek yang ada. Untuk informasi selengkapnya, lihat Paket ekstensi .
Microsoft SQL Server DW		
BigQuery	Amazon Redshift	Menyelesaikan masalah di mana objek yang difilter tidak dikonversi dalam mode antarmuka baris perintah (CLI).
Greenplum	Amazon Redshift	Memperbaiki kesalahan saat AWS SCT tidak mengonversi tabel sementara yang dideklarasikan dalam prosedur tersimpan.
Greenplum	Amazon Redshift	Memperbaiki kesalahan di mana atribut pengkodean kolom hilang dalam kode yang dikonversi.
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Konversi UPDATE pernyataan yang diterapkan untuk tabel referensi diri yang memiliki lebih dari satu INNER JOIN klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Menerapkan dukungan <code>inserted</code> dan tabel <code>deleted</code> sementara yang digunakan SQL Server untuk DML pemicu.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Peningkatan konversi tipe yang ditentukan pengguna dalam prosedur tersimpan yang dibuat dalam skema database yang berbeda. Menyelesaikan masalah di mana AWS SCT tidak dapat menemukan tipe data dan menampilkan item tindakan 9996.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana tanda kurung siku (<code>[]</code>) tiba-tiba muncul di sekitar nama objek database dalam kode yang dikonversi.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana <code>@@ROWCOUNT</code> fungsi salah dikonversi.
Microsoft SQL Server DW	Amazon Redshift	Menerapkan dukungan <code>geometry</code> dan tipe <code>geography</code> data.
Microsoft SQL Server DW	Amazon Redshift	Menerapkan dukungan MAX kata kunci dalam deklarasi tipe data dalam kode yang dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server DW	Amazon Redshift	Peningkatan konversi DATEADD fungsi.
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan konversi SQL kode dalam aplikasi Java dengan menambahkan dukungan untuk MyBatis kerangka kerja. Untuk informasi selengkapnya, lihat SQLkode di Jawa .
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan konversi SQL kode dalam aplikasi Java yang menggunakan MyBatis kerangka kerja. Menambahkan item tindakan 30411 untuk SQL kode dengan sintaks yang tidak didukung.
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan konversi SQL kode dalam aplikasi Pro* C dengan menambahkan dukungan untuk typedef struct deklarasi.
Oracle	Aurora PostgreSQL PostgreSQL	Menerapkan dukungan CROSS JOIN dan LEFT JOIN pernyataan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi MERGE pernyataan. Menyelesaikan masalah di mana nilai yang akan disisipkan tidak ada dalam kode yang dikonversi.
Teradata	Amazon Redshift	Mengubah setelan pengkodean kompresi kolom default yang AWS SCT digunakan dalam kode yang dikonversi agar sesuai dengan setelan Amazon Redshift default. Untuk informasi selengkapnya, lihat Pengkodean kompresi di Panduan Pengembang Database Amazon Redshift.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana operasi matematika yang menggunakan tipe TIME data salah dikonversi.
Teradata	Pergeseran Merah Amazon RSQL	Menerapkan konversi FastExport kode yang ada di dalam skrip shell.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Memperbaiki kesalahan di mana AWS SCT tidak dikonversi COALESCE dan %data pernyataan.
Vertica	Amazon Redshift	Saran optimasi konversi yang ditingkatkan saat pengguna memilih satu strategi pengoptimalan.

Catatan rilis untuk AWS SCT Build 666

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Mengatasi kesalahan penguraian yang terjadi untuk ON klausa yang ada di dalam pernyataan. JOIN
Analisis Sinaps Azure	Amazon Redshift	Menambahkan tiga fungsi baru ke paket ekstensi untuk meningkatkan konversi CONVERT fungsi dengan argumen tanggal dan waktu.
Analisis Sinaps Azure	Amazon Redshift	Meningkatkan pemuat metadata sumber untuk memastikan bahwa AWS SCT memuat skema database sistem.
Analisis Sinaps Azure	Amazon Redshift	Memperbaiki kesalahan resolver yang terjadi untuk kolom tabel sementara.
Analisis Sinaps Azure	Amazon Redshift	Menerapkan konversi BINARY dan tipe VARBINARY data ke tipe VARBYTE data.
Analisis Sinaps Azure	Amazon Redshift	Dukungan yang diimplementasikan dari tipe TIME data dalam kode yang dikonversi.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi COLLATE klausa. Menyelesaikan masalah saat item tindakan 31141 muncul secara tak terduga selama konversi kolom dengan pemeriksaan basis data default.
BigQuery	Amazon Redshift	Menerapkan konversi prosedur yang mengubah parameter input.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Greenplum	Amazon Redshift	Menyelesaikan masalah saat AWS SCT menggunakan kueri yang tidak kompatibel dengan database Greenplum 6.x.
IBMDb2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Peningkatan konversi EXCEPTION bagian dengan mentransfer penanganan pengecualian dari Db2 untuk z/OS ke PostgreSQL.
IBMDb2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Peningkatan konversi OPEN CURSOR pernyataan.
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Menerapkan konversi IIF fungsi dengan menggunakan CASE ekspresi.
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana prosedur dengan parameter bernilai tabel salah dikonversi saat CREATE PROCEDURE pernyataan tidak menyertakan BEGIN...END blok.
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana SCOPE_IDENTITY fungsi tersebut salah dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Amazon RDS untuk Oracle	Memperbaiki kesalahan loader yang terjadi dengan SELECT_CATALOG_ROLE peran saat menggunakan Oracle 10g sebagai sumber.
Oracle	Amazon RDS untuk Oracle	Meningkatkan loader untuk mendukung pekerjaan Oracle Scheduler.
Oracle	Aurora PostgreSQL PostgreSQL	Menerapkan konversi JOIN pernyataan dengan USING klausa.
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan kinerja kode yang dikonversi di mana kode sumber mencakup variabel global dalam WHERE klausa.
Oracle	Aurora PostgreSQL PostgreSQL	Peningkatan konversi SQL kode dalam aplikasi Java dengan menambahkan dukungan untuk MyBatis kerangka kerja. Untuk informasi selengkapnya, lihat SQLkode di Jawa .
Oracle DW	Amazon Redshift	Konversi yang diimplementasikan dari operator PIVOT dan UNPIVOT relasional.
Teradata	Amazon Redshift	Memperbaiki kesalahan di mana kode sumber yang menggunakan JSON objek tidak dikonversi.
Teradata	Amazon Redshift	Memperbaiki kesalahan di mana tabel yang dibuat oleh pengguna yang jatuh tidak dimuat dengan benar.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Menerapkan konversi INSTR fungsi ke fungsi Amazon Redshift STRPOS asli.
Teradata	Amazon Redshift	Konversi yang diimplementasikan dari NVP dan TRANSLATE fungsi.
Teradata	Amazon Redshift	Peningkatan konversi COALESCE ekspresi.
Teradata	Amazon Redshift	Peningkatan konversi DECLARE CONDITION pernyataan.
Teradata	Amazon Redshift	Peningkatan konversi EXTRACT fungsi dengan elemen SECOND sintaks.
Teradata	Amazon Redshift	Peningkatan konversi SQLSTATE dan SQLCODE variabel di dalam LOOP pernyataan.
Teradata	Amazon Redshift	Peningkatan konversi indeks unik.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana item tindakan 9996 tiba-tiba muncul selama konversi CURRENT_TIMESTAMP pernyataan dengan presisi fraksional disetel ke 3.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana garis miring terbalik salah dikonversi dalam literal string.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana EXEC pernyataan yang dikonversi menyertakan nama bidang yang salah dalam ADD CONSTRAINT pernyataan.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana QUALIFY subquery dikonversi termasuk nama subquery yang salah.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Menyelesaikan masalah saat tampilan yang dikonversi tidak diterapkan. Menambahkan cast eksplisit ke tipe data tertentu untuk NULL nilai dalam kode yang dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana fungsi tanggal dan waktu salah dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana literal string heksadesimal tidak dikonversi.

Catatan rilis untuk AWS SCT Build 665

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Analisis Sinaps Azure	Amazon Redshift	Konversi CONCAT fungsi yang diimplementasikan dengan VARCHAR argumen.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi CREATE TABLE pernyataan yang membuat tabel sementara dan tidak menyertakan nama skema. AWS SCT membuat dbo skema untuk menyimpan tabel sementara ini dalam database target.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi DROP TABLE pernyataan yang Anda jalankan pada tabel sementara.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi OBJECT_ID pernyataan dengan BEGIN...END blok.
Analisis Sinaps Azure	Amazon Redshift	Menyelesaikan kesalahan di mana AWS SCT tidak dapat mengonversi prosedur yang disimpan dengan komentar blok.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
BigQuery	Amazon Redshift	Menerapkan konversi gudang BigQuery data ke Amazon Redshift. Untuk informasi selengkapnya, lihat Menghubungkan ke Google BigQuery dengan AWS Schema Conversion Tool .
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Peningkatan konversi pemacu yang menangani beberapa peristiwa dan bekerja dengan <code>inserted</code> dan tabel <code>deleted</code> sistem di SQL Server.
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Memperbaiki kesalahan resolver yang terjadi untuk <code>inserted</code> dan tabel <code>deleted</code> sistem di SQL Server.
SQLServer Microsoft	Babelfish untuk Aurora PostgreSQL	Menerapkan dukungan versi baru 2.1.0 dari file konfigurasi fitur Babelfish. File ini mendefinisikan SQL fitur yang didukung dan tidak didukung oleh versi Babelfish tertentu.
Oracle	Aurora MySQL MariaDB MySQL	Menyelesaikan masalah di mana tipe <code>varchar2</code> data salah dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Saya SQL	Untuk database Oracle versi 12c dan yang lebih tinggi, AWS SCT mendukung tipe data tambahan berikut:
	Aurora Postgre SQL	<ul style="list-style-type: none"> • VARCHAR2 • NVARCHAR2 • RAW
	MariaDB Saya SQL Postgre SQL	AWS SCT meningkatkan panjang kolom maksimum yang didukung dari 8.000 menjadi 32.767 byte untuk tipe data ini.
Oracle	Aurora Postgre SQL Postgre SQL	Mengatasi kesalahan parsing yang terjadi untuk paket Oracle Event Processing.
Teradata	Amazon Redshift	Menambahkan item tindakan 13214 untuk beberapa RESET WHEN klausa dalam satu pernyataan. SELECT
Teradata	Amazon Redshift	Menambahkan item tindakan untuk SQLSTATE variabel yang terletak di luar blok penanganan pengecualian.
Teradata	Amazon Redshift	Konversi ACTIVITY_COUNT variabel yang diterapkan keROW_COUNT .
Teradata	Amazon Redshift	Konversi yang diterapkan dari ST_TRANSFORM fungsi geometri bawaan.
Teradata	Amazon Redshift	Peningkatan konversi pernyataan hapus dalam tampilan tanpa WHERE klausa.
Teradata	Amazon Redshift	Peningkatan konversi CAST operator dalam ekspresi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Peningkatan konversi GROUP BY klausa.
Teradata	Amazon Redshift	Peningkatan konversi INSTR dan fungsi REGEXP_INSTR bawaan.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana referensi alias kolom lateral salah dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana nama kolom salah dikonversi dalam QUALIFY subquery.
Teradata	Amazon Redshift	Menerapkan konversi .QUIT perintah dengan kata kunci nilai ERRORCODE status.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana item tindakan 9996 muncul secara tak terduga selama konversi pernyataan. CREATE
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana item tindakan 9998 muncul secara tak terduga selama konversi pernyataan. END

Catatan rilis untuk AWS SCT Build 664

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan dukungan Amazon Redshift Serverless sebagai sumber dan target untuk proyek migrasi database di AWS SCT Untuk terhubung ke Amazon Redshift Tanpa Server, pastikan Anda menggunakan driver Amazon Redshift versi 2.1.0.9 atau lebih tinggi JDBC.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Meningkatkan antarmuka pengguna jendela Pengaturan konversi. AWS SCT sekarang menampilkan pengaturan hanya untuk pasangan konversi database dengan aturan pemetaan yang dibuat. Untuk informasi selengkapnya, lihat Pemetaan tipe data .
Semua	Semua	Memperbarui laporan penilaian untuk menghapus informasi duplikat tentang garis dan posisi item tindakan.
Semua	Amazon Redshift	Menerapkan penyeimbangan memori otomatis dalam tugas ekstraksi data.
Semua	Amazon Redshift	Mengatasi kesalahan di mana agen ekstraksi data tidak dapat terhubung ke AWS Snowball perangkat.
Database Azure SQL	Aurora Saya SQL	Menerapkan dukungan SUSE Linux 15.3 sebagai platform untuk menjalankan agen ekstraksi data.
IBMDB2 untuk z/OS	Aurora Postgre SQL	
IBMDB2 LUW	MariaDB	
SQLServer Microsoft	Saya SQL	
Saya SQL	Postgre SQL	
Oracle		
Postgre SQL		
SAP ASE		
Analisis Sinaps Azure	Amazon Redshift	

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
IBMDB2 untuk z/OS	Aurora Postgre SQL Postgre SQL	Menambahkan kemampuan untuk mengubah pemeriksaan kolom dalam aturan migrasi.
Microsoft SSIS	AWS Glue AWS Glue Studio	Mengatasi kesalahan tak terduga yang terjadi saat pengguna memilih skrip sumber.
Oracle	Aurora Saya SQL MariaDB Saya SQL	Konversi yang diimplementasikan dari penggunaan fungsi yang disimpan sebagai ekspresi kolom yang dihasilkan. AWS SCT membuat pemacu untuk meniru perilaku ini karena My SQL tidak mendukung penggunaan fungsi yang disimpan sebagai ekspresi kolom yang dihasilkan.
Oracle	Aurora Postgre SQL Postgre SQL	Menerapkan konversi fungsi dari UTL_MATCH paket sebagai bagian dari paket AWS SCT ekstensi.
Oracle	Aurora Postgre SQL Postgre SQL	Konversi REGEXP_LIKE fungsi yang diimplementasikan dengan NULL parameter.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi SYS_EXTRACT_UTC fungsi.
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi SQL kode dalam aplikasi C ++ dengan menerapkan dukungan Wcscats, Wcscpys, dan Wcsncats fungsi. Untuk informasi selengkapnya, lihat Mengonversi SQL kode dalam aplikasi C ++ dengan AWS Schema Conversion Tool .
Oracle DW Kepingan salju	Amazon Redshift	Menyelesaikan masalah di mana pernyataan yang dikonversi tidak menyertakan konversi nilai eksplisit ke tipe data kolom. Masalah ini terjadi dalam pernyataan yang menggunakan hasil kueri dari tabel lain.
Teradata	Amazon Redshift	Menambahkan kemampuan untuk mengubah pemeriksaan kolom antara case sensitive dan case insensitive dalam aturan migrasi. Untuk informasi selengkapnya, lihat Menerapkan aturan migrasi .
Teradata	Amazon Redshift	Memperbaiki kesalahan resolver yang terjadi untuk CREATE TABLE AS pernyataan.
Teradata	Amazon Redshift	Memperbaiki kesalahan di mana P_INTERSECT fungsi bawaan dengan COALESCE ekspresi tidak dikonversi.
Teradata	Amazon Redshift	Menerapkan konversi kolom bernama OID _OID untuk menghindari penggunaan kata kunci yang dicadangkan di Amazon Redshift.
Teradata	Amazon Redshift	Menerapkan konversi RENAME pernyataan untuk fungsi, prosedur, tampilan, dan makro.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Menerapkan konversi <code>STROKE</code> fungsi ke <code>SPLIT_PART</code> fungsi di Amazon Redshift.
Teradata	Amazon Redshift	Peningkatan konversi fungsi <code>INSTR</code> dan <code>REGEXP_INSTR</code> sistem.
Teradata	Amazon Redshift	Peningkatan konversi tipe <code>TIME</code> data.
Teradata	Amazon Redshift	Peningkatan emulasi <code>SET</code> dan <code>MULTISET</code> tabel dengan menerapkan konversi indeks unik primer dan sekunder.
Teradata	Amazon Redshift	Menyelesaikan kesalahan penguraian yang terjadi untuk <code>CHARACTER</code> fungsi tersebut.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Mengatasi kesalahan yang terjadi saat pengguna menghapus skrip Teradata Basic Teradata Query (BTEQ) dari proyek. AWS SCT

Catatan rilis untuk AWS SCT Build 663

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan kemampuan untuk mengubah <code>panjangchar</code> , <code>varcharnvarchar</code> , dan tipe <code>string</code> data menggunakan operator perkalian dalam aturan migrasi. Untuk informasi selengkapnya, lihat Menerapkan aturan migrasi .
Semua	Semua	Menerapkan dukungan tiga kolom baru dalam laporan penilaian multiserver dan memperbarui format file input. Pastikan Anda menggunakan template yang diperbarui dari file input dengan versi terbaru AWS SCT. Untuk informasi selengkapnya, lihat Membuat laporan penilaian multiserver di AWS Schema Conversion Tool .

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi OBJECT_ID pernyataan.
SQLServer Microsoft	Babelfish untuk Aurora Postgre SQL	Menambahkan dukungan untuk Babelfish untuk Aurora Postgre SQL 1.2.0 sebagai platform target untuk laporan penilaian migrasi database. Untuk informasi selengkapnya, lihat Fungsionalitas yang didukung di Babelfish menurut versi di Panduan Pengguna Amazon Aurora.
Microsoft SQL Server DW	Amazon Redshift	Menambahkan dukungan untuk AT TIME ZONE klausa.
Microsoft SQL Server DW	Amazon Redshift	Menyelesaikan masalah di mana pernyataan di luar BEGIN/END blok salah dikonversi.
Netezza	Amazon Redshift	Peningkatan konversi tipe TIME data dan konversi yang diimplementasikan dari fungsi, ekspresi, dan literal bawaan terkait.
Oracle	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan loader yang terjadi saat menggunakan Oracle 10g sebagai sumber.
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi OFFSET dan FETCH klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana prosedur dengan OUT parameter dengan nilai default salah dikonversi.
Oracle DW	Amazon Redshift	Peningkatan konversi fungsi Oracle ke fungsi yang ditentukan pengguna Amazon Redshift.
Kepingan salju	Amazon Redshift	Peningkatan konversi WITH klausa.
Teradata	Amazon Redshift	Menambahkan item tindakan baru 13209 untuk karakter multibyte yang tidak didukung untuk tipe data. CHAR
Teradata	Amazon Redshift	Memperbaiki kesalahan pemuat di mana tabel tidak terisi penuh.
Teradata	Amazon Redshift	Memperbaiki kesalahan transformator di mana P_INTERSECT fungsi bawaan dalam suatu JOIN kondisi tidak dikonversi.
Teradata	Amazon Redshift	Memperbaiki masalah di mana nama tampilan diubah dalam kasus yang salah ketika SELECT pernyataan dijalankan di atas meja dengan karakter khusus dalam namanya.
Teradata	Amazon Redshift	Peningkatan konversi INSERT pernyataan dengan UNTIL_CHANGED nilai dalam tipe PERIOD(DATE) data.
Teradata	Amazon Redshift	Peningkatan konversi fungsi bawaan menggunakan FORMAT TO_CHAR fungsi di Amazon Redshift.
Teradata	Amazon Redshift	Peningkatan konversi RANK fungsi bawaan untuk memastikan bahwa kode yang dikonversi mengembalikan NULL nilai dalam urutan yang sama dengan kode sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Peningkatan konversi kendala unik seperti indeks unik primer atau sekunder.

Catatan rilis untuk AWS SCT Build 662

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan kemampuan untuk secara otomatis membuat AWS SCT proyek untuk setiap database sumber saat membuat laporan penilaian multiserver. Dengan opsi ini diaktifkan, AWS SCT dapat menambahkan aturan pemetaan untuk proyek-proyek ini dan menyimpan statistik konversi untuk penggunaan offline. Untuk informasi selengkapnya, lihat Membuat laporan penilaian multiserver di AWS Schema Conversion Tool .
Semua	Semua	Menerapkan dukungan persen (%) sebagai wildcard dalam database dan nama skema saat membuat laporan penilaian multiserver.
Semua	Aurora Saya SQL Aurora Postgre SQL	Memperbarui runtime semua AWS Lambda fungsi ke Python versi 3.9.
Semua	Amazon Redshift	Upgrade semua agen ekstraksi data untuk digunakan AWS SDK for Java 2.x.
Database Azure SQL SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Peningkatan konversi DELETE pernyataan dengan NON EXISTS klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Analisis Sinaps Azure	Amazon Redshift	Menyelesaikan kesalahan di mana koneksi ke database sumber gagal.
IBMDB2 untuk z/OS	Aurora PostgreSQL PostgreSQL	Menyelesaikan kesalahan di mana kode dikonversi dari pemacu termasuk dua penyebutan alias objek.
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Peningkatan konversi objek dengan nama dalam kasus campuran ketika Treat nama objek database sebagai opsi case sensitive dihidupkan.
Microsoft SQL Server DW Teradata	Amazon Redshift	Konversi yang diimplementasikan dari operator PIVOT dan UNPIVOT relasional.
Netezza	Amazon Redshift	Konversi yang diimplementasikan dari tipe TIME data.
Oracle	Aurora Saya SQL Aurora PostgreSQL Saya SQL PostgreSQL	Menerapkan UTL_TCP.CRLF paket konversi konstan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Memperbaiki masalah paket ekstensi di mana panjang tipe data untuk kolom dengan panjang variabel tidak dipertahankan selama konversi.
Oracle	Aurora Postgre SQL Postgre SQL	Konversi SQL kode yang diterapkan dalam aplikasi C ++. Untuk informasi selengkapnya, lihat Mengonversi SQL kode dalam aplikasi C ++ dengan AWS Schema Conversion Tool .
Oracle	Aurora Postgre SQL Postgre SQL	Menerapkan dukungan penamaan case sensitive untuk konversi variabel global dan array asosiatif.
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi TO_CHAR, TO_DATE, dan TO_NUMBER fungsi dalam paket ekstensi.
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi TABLE () operator.
Oracle DW	Amazon Redshift	Ditambahkan dukungan untuk konversi kunci utama dan kendala lainnya.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle DW	Amazon Redshift	Memperbaiki masalah saat item tindakan 12054 tidak muncul selama konversi pernyataan bersyarat.
SAP ASE	Aurora PostgreSQL PostgreSQL	Menyelesaikan kesalahan saat objek dengan nama kosong dibuat di pohon target selama konversi tabel dengan kolom tipe yang ditentukan pengguna.
SAP ASE	Aurora PostgreSQL PostgreSQL	Memperbaiki kesalahan loader untuk objek yang disimpan seperti skrip, rutinitas, dan sebagainya.
Kepingan salju	Amazon Redshift	Memperbaiki masalah saat item tindakan 22152 tidak muncul saat diperlukan dan AWS SCT menampilkan hasil konversi sebagai komentar.
Kepingan salju	Amazon Redshift	Peningkatan konversi fungsi tanggal dan waktu; menerapkan dukungan zona waktu.
Kepingan salju	Amazon Redshift	Menyelesaikan masalah di mana ekspresi tabel umum non-rekursif (CTEs) dengan WITH klausa diubah sebagai rekursif. CTEs
Teradata	Amazon Redshift	Peningkatan konversi UPDATE pernyataan dengan tautan tabel dalam kondisi.
Teradata	Amazon Redshift	Peningkatan konversi RENAME TABLE pernyataan.
Teradata	Amazon Redshift	Menyelesaikan masalah saat kolom kosong muncul di file value (CSV) yang dipisahkan koma dengan laporan penilaian.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Pergeseran Merah Amazon RSQL	Memperbaiki kesalahan saat titik koma hilang di akhir makro Basic Teradata Query () yang dikonversi. BTEQ
Teradata	Pergeseran Merah Amazon RSQL	Peningkatan konversi beberapa nilai tipe data dalam CASE pernyataan.
Teradata	Pergeseran Merah Amazon RSQL	Peningkatan konversi LIKE ANY klausa dengan ESCAPE karakter.
Teradata	Pergeseran Merah Amazon RSQL	Peningkatan konversi CAST fungsi dalam INSERT pernyataan.
Teradata	Pergeseran Merah Amazon RSQL	Peningkatan konversi zona waktu, menerapkan pemetaan wilayah zona waktu.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana item tindakan 9998 tiba-tiba muncul selama konversi skrip shell dengan skrip. BTEQ
Teradata	Pergeseran Merah Amazon RSQL AWS Glue	Menerapkan batas 500 karakter untuk nilai-nilai variabel substitusi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Vertica	Amazon Redshift	Konversi yang diimplementasikan dari BINARY VARBINARY LONG BINARY,BYTEA,,, dan tipe RAW data ke tipe VARBYTE data.
Vertica	Amazon Redshift	Peningkatan konversi fungsi dan literal bawaan.

Catatan rilis untuk AWS SCT Build 661

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan filter untuk mencari aturan pemetaan dalam tampilan pemetaan. Saat Anda menerapkan filter, AWS SCT menampilkan aturan yang cocok dengan kondisi pemfilteran dalam daftar pemetaan Server. Untuk informasi selengkapnya, lihat Mengedit pemetaan tipe data di AWS Schema Conversion Tool .
Semua	Semua	Upgrade Apache Log4j ke versi 2.17.1.
Semua	Amazon Redshift	Menambahkan dukungan migrasi data ke Amazon Redshift menggunakan ENCRYPTED klausa dalam perintah. COPY
Semua	Amazon Redshift	Meningkatkan REST API agen ekstraksi data. Yang diperbarui REST API menambahkan dukungan properti baru seperti kunci enkripsi, jenis enkripsi, dan sebagainya.
Semua	Amazon Redshift	Menerapkan peran dengan asumsi dalam agen ekstraksi data. Pembaruan ini meningkatkan distribusi subtugas, dan memungkinkan AWS SCT untuk menetapkan tugas ke agen bebas dari peran yang ditentukan.
Semua	Amazon Redshift	Menerapkan pemeriksaan bahwa semua komponen yang diperlukan diinstal sebelum paket ekstensi diterapkan ke Amazon Redshift.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Analisis Sinaps Azure	Amazon Redshift	Peningkatan konversi fungsi <code>ERROR_LINE</code> , <code>ERROR_MESSAGE</code> , <code>ERROR_NUMBER</code> <code>ERROR_PROCEDURE</code> , <code>ERROR_SEVERITY</code> , dan <code>ERROR_STATE</code> sistem untuk penanganan kesalahan.
Microsoft SQL Server DW		
IBMDB2 untuk z/OS	Aurora Saya SQL Aurora Postgre SQL Saya SQL Postgre SQL	Menambahkan dukungan IBM Db2 untuk z/OS versi 12 sebagai sumber untuk proyek migrasi database di. AWS SCT Untuk informasi selengkapnya, lihat Menghubungkan ke IBM DB2 untuk z/OS .
IBMDB2 LUW	Semua	Meningkatkan pemuat metadata sumber untuk memastikan bahwa AWS SCT memuat parameter rutin yang menduplikasi nama kolom.
Database Microsoft Azure SQL	Aurora Postgre SQL	Memperbaiki kesalahan transformator untuk prosedur dengan pernyataan yang <code>SET NOCOUNT ON</code> ditetapkan.
SQLServer Microsoft	Postgre SQL	
Database Microsoft Azure SQL	Aurora Postgre SQL	Peningkatan konversi <code>CONCAT</code> fungsi ketika nilai input adalah variabel dari tipe yang ditentukan pengguna.
SQLServer Microsoft	Postgre SQL	

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Database Microsoft Azure SQL SQLServer Microsoft	Aurora Postgre SQL	Menyelesaikan masalah di mana DATEPART fungsi tersebut salah dikonversi.
SQLServer Microsoft	Babelfish untuk Aurora Postgre SQL	Menerapkan dukungan versi baru dari file konfigurasi fitur Babelfish . File ini mendefinisikan SQL fitur yang didukung dan tidak didukung oleh versi Babelfish tertentu.
Microsoft SQL Server DW	Amazon Redshift	Menyelesaikan masalah di mana prosedur dengan EXECUTE pernyataan salah dikonversi.
Microsoft SSIS	AWS Glue	Meningkatkan antarmuka pengguna wizard konfigurasi pekerjaan. AWS SCT sekarang hanya menampilkan koneksi yang tersedia di bagian konfigurasi koneksi.
Microsoft SSIS	AWS Glue	Menyelesaikan masalah di mana aturan transformasi tidak diterapkan pada tugas paket dan aturan variabel.
Microsoft SSIS	AWS Glue AWS Glue Studio	Menambahkan item tindakan baru 25042 untuk komponen yang tidak didukung.
Microsoft SSIS	AWS Glue Studio	Konversi yang diimplementasikan dari Microsoft SQL Server Integration Services (SSIS) mengekstrak, mengubah, dan memuat (ETL) paket ke AWS Glue Studio. Untuk informasi selengkapnya, lihat SSIS untuk AWS Glue Studio .
Oracle	MariaDB	Memperbaiki masalah dengan konversi MINUS operator.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	MariaDB	Peningkatan konversi dari ROWNUM, SYS_GUIDTO_CHAR, dan ADD_MONTHS fungsi ketika variabel sql_mode sistem di MariaDB adalah untuk Oracle.
Oracle	Postgre SQL	Menambahkan opsi untuk menghindari konversi jenis variabel bind ke SQL jenis dalam proyek konversi aplikasi generik.
Oracle	Postgre SQL	Menambahkan opsi untuk menghindari menambahkan nama skema ke nama objek yang dikonversi dalam proyek konversi aplikasi generik.
Oracle	Postgre SQL	Ditambahkan dukungan dari format variabel ?x mengikat untuk konversi SQL kode aplikasi.
Oracle DW	Amazon Redshift	Konversi yang diimplementasikan dari tipe RAW data ke tipe VARBYTE data.
Teradata	Amazon Redshift	Ditambahkan pilihan untuk meniru SET tabel dalam kode dikonversi. Untuk emulasi ini, AWS SCT dukungan MIN dan MAX kondisi.
Teradata	Amazon Redshift	Peningkatan konversi operasi gabungan yang memiliki parameter tipe data yang berbeda. Pembaruan ini memungkinkan AWS SCT untuk menerapkan aturan transformasi selama konversi operasi tersebut.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana GROUP BY klausa salah dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana QUALIFY klausa salah dikonversi.
Teradata	Amazon Redshift	Teratasi kesalahan tak terduga terjadi selama impor FastExport skrip.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Pergeseran Merah Amazon RSQL	Menerapkan kemampuan untuk mengedit nilai-nilai variabel dalam Teradata BTEQ dan skrip shell.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana skrip manifes tidak ada untuk sesi Teradata FastLoad yang dikonversi.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana ekstensi file manifes hilang di uniform resource locator (URL) untuk FastLoad skrip yang dikonversi.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Memperbaiki kesalahan loader untuk skrip dengan variabel substitusi.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Memperbaiki masalah saat item tindakan 27022 tidak muncul saat diperlukan.

Catatan rilis untuk AWS SCT Build 660

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan dukungan AWS Secrets Manager dan Secure Sockets Layer (SSL) dalam laporan penilaian multiserver. Untuk informasi selengkapnya, lihat Membuat laporan penilaian multiserver di AWS Schema Conversion Tool .
Semua	Semua	Peningkatan koleksi statistik untuk objek dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Postgre SQL	Menerapkan dukungan Postgre SQL mayor versi 14 dan MariaDB 10.6 sebagai target migrasi.
Analisis Sinaps Azure	Amazon Redshift	Peningkatan logika transformasi untuk nama-nama objek yang dikonversi.
Database Microsoft Azure SQL Server Microsoft	Aurora Postgre SQL	Peningkatan konversi tipe XML data.
Database Microsoft Azure SQL Server Microsoft	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana NOT LIKE klausa salah dikonversi.
Database Microsoft Azure SQL Server Microsoft	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan transformator untuk prosedur dengan INSERT, DELETE, dan UPDATE pernyataan yang menyertakan OUTPUT klausa.
Database Microsoft Azure SQL Server Microsoft	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan transformator untuk prosedur dengan RETURN @@ROWCOUNT pernyataan.
SQLServer Microsoft	Semua	Peningkatan konversi prosedur yang menggunakan server tertaut.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	Semua	Menambahkan dukungan Microsoft Windows Authentication dalam laporan penilaian multiserver.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Memperbaiki kesalahan transformator untuk konstruktor nilai tabel.
Microsoft SQL Server DW	Amazon Redshift dan AWS Glue	Peningkatan konversi skrip ekstrak, transformasi, dan load (ETL) untuk menyertakan jalur yang benar ke skrip yang dikonversi.
Microsoft SQL Server DW	Amazon Redshift	Menyelesaikan masalah di mana skrip yang dikonversi yang berbeda dibuat untuk platform basis data target virtual dan nyata.
Oracle	Postgre SQL Aurora Postgre SQL	Menambahkan dukungan untuk konversi indeks untuk tampilan terwujud.
Oracle	Postgre SQL Aurora Postgre SQL	Memperbaiki masalah saat item tindakan 5982 tidak muncul saat mengonversi PRIMARY KEY dan UNIQUE membatasi opsi. NOVALIDATE
Oracle DW	Amazon Redshift	Menyelesaikan masalah di mana kategori tambahan ditampilkan dalam skema yang dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Memperbaiki masalah saat item tindakan 13185 tidak muncul saat mengonversi kolom yang belum terselesaikan sebagai argumen fungsi. CAST
Teradata	Amazon Redshift	Peningkatan konversi DELETE dan DELETE ALL pernyataan untuk menggunakan TRUNCATE perintah dalam kode yang dikonversi.
Teradata	Amazon Redshift	Peningkatan konversi SET tabel.
Teradata	Amazon Redshift	Peningkatan konversi NORMALIZE kondisi.
Teradata	Amazon Redshift	Memperbarui laporan penilaian untuk menghapus statistik konversi skema database dari daftar objek penyimpanan database.
Teradata	Amazon Redshift	Peningkatan konversi UPDATE pernyataan tanpa FROM klausa.
Teradata	Amazon Redshift	Dukungan yang diimplementasikan dari tipe VARBYTE data dalam kode yang dikonversi.
Teradata BTEQ	AWS Glue	Menyelesaikan masalah saat AWS Glue opsi Convert to dinonaktifkan di menu konteks.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana tipe data hilang dalam kode yang dikonversi.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana variabel substitusi salah dikutip dalam kode yang dikonversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Memperbaiki masalah dengan konversi variabel substitusi dengan nilai dalam FastLoad skrip.
Vertica	Amazon Redshift	Dukungan yang diimplementasikan dari tipe TIME data dalam kode yang dikonversi.
Vertica	Amazon Redshift	Peningkatan konversi SELECT DISTINCT dan ORDER BY ekspresi.
Vertica	Amazon Redshift	Ditambahkan dukungan untuk konversi kendala.
Vertica	Amazon Redshift	Mengatasi kesalahan saat laporan penilaian tidak disimpan sebagai file value (CSV) yang dipisahkan koma.

Catatan rilis untuk AWS SCT Build 659

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Memperbaiki wizard proyek Baru yang menghasilkan laporan penilaian gabungan untuk beberapa basis data sumber.
Semua	Semua	Memperbaiki masalah saat paket ekstensi tidak dibuat dalam proyek yang menyertakan beberapa basis data sumber dan target.
Semua	Semua	Peningkatan konversi SQL kode yang tertanam dalam kode sumber aplikasi.
Semua	Semua	Menambahkan kemampuan untuk menjalankan skrip dari folder yang berbeda di antarmuka baris AWS SCT perintah.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Amazon Redshift	Memperbaiki pesan peringatan yang diberikan saat pengguna memilih Jalankan pengoptimalan dalam proyek migrasi dengan platform basis data target virtual Amazon Redshift.
Semua	Aurora PostgreSQL	Menerapkan dukungan PostgreSQL mayor versi 13 pada Aurora PostgreSQL -Compatible Edition sebagai target migrasi.
Semua	Amazon RDS untuk MySQL	Menerapkan konversi kode case insensitive secara default.
Analisis Sinaps Azure	Amazon Redshift	Mengatasi kesalahan di mana koneksi ke database sumber gagal di antarmuka baris perintah.
SQLServer Microsoft	PostgreSQL Aurora PostgreSQL	Peningkatan konversi prosedur yang mencakup UPDATE pernyataan dengan kondisi gabungan.
SQLServer Microsoft	PostgreSQL Aurora PostgreSQL	Peningkatan konversi pemacu, prosedur tersimpan, dan fungsi yang mencakup nilai setelah tanda sama dengan.
SQLServer Microsoft	PostgreSQL Aurora PostgreSQL	Memperbaiki kesalahan transformator untuk prosedur dengan DELETE pernyataan dan OR operator.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	Postgre SQL Aurora Postgre SQL	Peningkatan konversi OUTPUT klausa.
Microsoft SQL Server DW	Amazon Redshift dan AWS Glue	Peningkatan konversi tipe NUMERIC data.
Microsoft SQL Server DW	Amazon Redshift	Peningkatan konversi tampilan yang memiliki alias tabel dengan nama yang sama dengan tabel asli.
Microsoft SSIS	AWS Glue	Memperbaiki masalah saat kredensi AWS Glue koneksi tidak ditampilkan di jendela Konfigurasi koneksi.
Netezza	Amazon Redshift	Menambahkan kemampuan untuk mengulangi proses perubahan data capture (CDC) tugas migrasi data setiap hari.
Netezza	Amazon Redshift	Memperbaiki masalah saat tab Tugas menjadi tidak aktif setelah membatalkan pendaftaran agen ekstraksi data.
Netezza	Amazon Redshift	Memperbaiki masalah saat konfirmasi pendaftaran agen migrasi data tidak ditampilkan di antarmuka pengguna.
Netezza	Amazon Redshift	Memperbaiki masalah saat koneksi ke database sumber gagal dengan kesalahan Loader.
Netezza	Amazon Redshift	Mengatasi kesalahan di mana agen migrasi data gagal dijalankan setelah membuka proyek yang disimpan.
Oracle	Amazon RDS untuk Oracle	Menerapkan dukungan Oracle Unified Audit.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Postgre SQL Aurora Postgre SQL	Konversi SQL kode yang diterapkan dalam aplikasi C #. Untuk informasi selengkapnya, lihat SQLkode dalam aplikasi C # .
Oracle	Postgre SQL Aurora Postgre SQL	Menerapkan logika transformasi baru untuk nama objek peka huruf besar/kecil untuk meningkatkan visibilitas perubahan konversi kode. AWS SCT mengubah nama objek dalam huruf besar menjadi huruf kecil. Kebalikannya juga benar; AWS SCT mengonversi nama objek dalam huruf kecil menjadi huruf besar. Nama objek lain dan kata-kata yang dipesan dikonversi tanpa perubahan.
Oracle	Postgre SQL Aurora Postgre SQL	Peningkatan konversi partisi hash tanpa kendala. NOT NULL
Oracle	Aurora Postgre SQL	Ditambahkan dukungan untuk konversi OracleCHECK, FOREIGN KEY, dan NOT NULL kendala dengan klausa. ENABLE NOVALIDATE
Oracle DW	Amazon Redshift	Memperbaiki masalah di mana nilai yang salah untuk nomor floating point dimigrasikan.
Oracle DW	Amazon Redshift dan AWS Glue	Menyelesaikan masalah dengan kolom kosong dalam laporan penilaian migrasi database dalam file value (CSV) yang dipisahkan koma.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SAP ASE	Postgre SQL Aurora PostgreSQL	Memperbaiki masalah dengan gangguan konversi yang tidak terduga.
Kepingan salju	Amazon Redshift	Peningkatan konversi tipe VARIANT data.
Teradata	Amazon Redshift	Peningkatan konversi COLLECT STATISTICS pernyataan.
Teradata	Amazon Redshift	Memperbaiki masalah saat item tindakan 9998 tidak muncul saat mengonversi tampilan bersarang dengan kolom. PERIOD
Teradata	Amazon Redshift dan AWS Glue	Memperbaiki masalah saat platform AWS Glue target virtual tidak ditampilkan di UI setelah membuka proyek yang disimpan.
Teradata BTEQ	AWS Glue	Memperbaiki masalah saat konversi ke platform AWS Glue target virtual tidak didukung setelah membuka proyek yang disimpan.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Penyorotan sintaks yang ditingkatkan dari kode yang dikonversi.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Diimplementasikan memeriksa nilai parameter setelah upload. Nilai yang tidak didukung disorot pada tab Variabel.
Vertica	Amazon Redshift	Konversi fungsi agregat yang diimplementasikan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Vertica	Amazon Redshift	Menerapkan konversi proyeksi ke tampilan terwujud dan meningkatkan UI yang menampilkan kode sumber proyeksi.

Catatan rilis untuk AWS SCT Build 658

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menyediakan integrasi dengan AWS Secrets Manager. Anda sekarang dapat menggunakan kredensial koneksi database yang disimpan di Secrets Manager.
Semua	Semua	Menambahkan dukungan untuk skrip dalam YAML format di antarmuka baris AWS SCT perintah.
Semua	Amazon Redshift	Menerapkan dukungan titik akhir antarmuka Amazon S3 (VPCE) di agen ekstraksi data.
Semua	Amazon Redshift	Menambahkan dukungan untuk platform basis data target virtual Amazon Redshift selain Amazon AWS Glue Redshift dan kombinasi yang sudah didukung.
Greenplum	Amazon Redshift	Memperbaiki masalah di mana SQL opsi Simpan sebagai tidak menyimpan SQL kode yang dikonversi ke file.
IBMDb2 LUW Database Microsoft Azure SQL SQLServer Microsoft Oracle	Aurora Saya SQL	Peningkatan konversi untuk mendukung fitur baru Amazon Aurora My SQL -Compatible Edition dengan kompatibilitas My SQL 8.0.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SAP ASE		
SQLServer Microsoft	Aurora Saya SQL Aurora Postgre SQL Saya SQL Postgre SQL	Memperbaiki masalah saat item tindakan 810 tidak muncul saat diperlukan.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Peningkatan konversi prosedur dengan UPDATE, DELETE, dan INSERT pernyataan.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Memperbaiki masalah saat item tindakan 7810 tidak muncul saat diperlukan.
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Peningkatan konversi EXEC pernyataan yang bersarang di dalam IF...ELSE pernyataan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	Aurora Postgre SQL Postgre SQL	Peningkatan konversi tampilan yang diindeks.
Netezza	Amazon Redshift	Agan migrasi data yang ditingkatkan dengan melacak transaksi langsung selama pemuatan penuh dalam operasi change data capture (CDC). Anda sekarang dapat menghentikan tugas migrasi data jika CDC sesi dijadwalkan untuk dimulai pada waktu tertentu. Selain itu, Anda dapat melihat tingkat pencatatan kesalahan di konsol setelah Anda menghentikan tugasCDC.
Oracle	Semua	Meningkatkan pemuat tabel untuk memastikan bahwa AWS SCT memuat objek dengan opsi berbagi.
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi SYSDATE fungsi dan menambahkan kemampuan untuk mengubah zona waktu dalam pengaturan Konversi.
Oracle	Aurora Postgre SQL Postgre SQL	Menyelesaikan masalah di mana pernyataan dinamis tidak dikonversi.
Oracle	Aurora Postgre SQL Postgre SQL	Memperbaiki masalah saat kode yang dikonversi tidak menyertakan nama yang dihasilkan sistem.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle Oracle DW	Aurora Postgre SQL Postgre SQL	Peningkatan konversi SELECT pernyataan yang bersarang di dalam pemicu.
Oracle DW	Amazon Redshift	Peningkatan konversi TO_DATE, TO_TIMESTAMP , dan TO_TIMESTAMP_TZ fungsi dalam paket ekstensi.
Kepingan salju	Amazon Redshift	Ditambahkan pilihan untuk menyimpan SQL kode dikonversi dalam file yang berbeda untuk setiap objek atau untuk setiap pernyataan.
Teradata	Amazon Redshift	Peningkatan konversi CONCAT fungsi.
Teradata	Amazon Redshift	Peningkatan konversi SELECT pernyataan yang bersarang di dalam WHERE klausa.
Teradata	Amazon Redshift	Menyelesaikan masalah saat SET dan MULTiset tabel salah dikonversi setelah pengguna menjatuhkan dan membuat ulang tabel.
Teradata	Amazon Redshift	Peningkatan konversi prosedur yang mencakup WITH klausa.
Teradata	Amazon Redshift	Peningkatan konversi tipe DATE data.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah saat terjadi kesalahan transformator tak terduga selama konversi FastExport skrip.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Menambahkan dukungan untuk konversi indeks gabungan ke tampilan terwujud.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Menambahkan dukungan untuk konversi TITLE definisi yang mencakup beberapa baris.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana ukuran tipe data geospasial tidak dikonversi.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Memperbaiki masalah di mana nama parameter dikonversi ke karakter huruf kecil.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Memperbaiki masalah saat prosedur tersimpan yang bersarang di dalam MACRO pernyataan tidak dikonversi.
Vertica	Amazon Redshift	Peningkatan konversi ALL operator.
Vertica	Amazon Redshift	Menyelesaikan masalah di mana Use Union all view? opsi dalam pengaturan Konversi tidak diterapkan.
Vertica	Amazon Redshift	Peningkatan konversi dari TIME dan tipe TIME WITH TIMEZONE data.
Vertica	Amazon Redshift	Menyelesaikan masalah dengan pemuatan tabel fleksibel.

Masalah diselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 657

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Upgrade Apache Log4j ke versi 2.17 untuk mengatasi masalah kerentanan keamanan.
Semua	Amazon Redshift	Proyek pengoptimalan skema yang ditingkatkan, di mana statistik manajemen kunci tidak disimpan dalam proyek. AWS SCT
Amazon Redshift	Amazon Redshift	Memperbaiki masalah dengan pembaruan informasi server.
Apache Cassandra	Amazon DynamoDB	Memperbaiki masalah dengan aturan pemetaan saat menggunakan antarmuka baris AWS SCT perintah.
Apache Cassandra	Amazon DynamoDB	Menyelesaikan masalah saat tugas migrasi tidak dibuat karena judul yang diperbarui dalam sertifikat.
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Memperbaiki masalah sehingga item tindakan 7672 tidak muncul selama konversi prosedur Microsoft SQL Server dengan dinamisSQL.
Database Azure SQL SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Peningkatan konversi fungsi bernilai tabel.
Database Azure SQL SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah di mana OUT argumen dalam prosedur tersimpan dengan nilai pengembalian default tidak dikonversi ke INOUT argumen.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Greenplum	Amazon Redshift	Meningkatkan strategi optimasi dengan menemukan tabel dan kolom yang paling banyak digunakan dari QueryLog tabel.
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Memperbaiki masalah dengan konversi berikut ini: <ul style="list-style-type: none"> • Operator penugasan penggabungan string () += • Fungsi SCOPE_IDENTITY • Tipe data varchar(max)
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Peningkatan konversi tampilan dengan fungsi yang tidak didukung.
SQLServer Microsoft	Aurora PostgreSQL PostgreSQL	Memperbaiki masalah di mana fungsi yang tidak didukung sebagai argumen ke fungsi lain salah dikonversi.
SQLServer Microsoft	Babelfish untuk Aurora PostgreSQL	Peningkatan konversi referensi tabel transisi.
Microsoft SQL Server DW	Amazon Redshift	Menambahkan kategori fungsi agregat ke pohon metadata database sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server DW	Amazon Redshift	Peningkatan konversi tipe TIME data.
Analisis Sinaps Azure Greenplum Netezza Microsoft SQL Server DW Kepingan salju Teradata	Amazon Redshift	Memperbaiki masalah di mana DROP dan CREATE skrip tidak disimpan saat menggunakan platform basis data target virtual.
Layanan Integrasi Microsoft SQL Server	AWS Glue	Menyelesaikan masalah di mana skrip objek sumber tidak ditampilkan di UI.
Netezza	Amazon Redshift	Meningkatkan strategi optimasi dengan memilih tabel fakta dan dimensi yang sesuai untuk kolokasi.
Oracle	Aurora PostgreSQL PostgreSQL	Menyelesaikan masalah untuk mengonversi pemicu Oracle dengan benar, yang menggunakan nomor urut.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Aurora Postgre SQL Postgre SQL	Peningkatan konversi tampilan dengan link database publik.
Oracle DW	Amazon Redshift	Meningkatkan strategi optimasi dengan menganalisis kardinalitas kolom indeks.
Oracle DW	Amazon Redshift	Memperbaiki masalah saat fungsi skalar yang ditentukan pengguna khusus dengan rangkaian string tidak dikonversi dengan benar.
Kepingan salju	Amazon Redshift	Memperbaiki masalah saat SQL opsi Simpan sebagai tidak ditampilkan di UI.
Teradata	Amazon Redshift	Memperbaiki masalah saat pengumpulan statistik gagal dengan LOADER ERROR pengecualian.
Teradata	Amazon Redshift	Memperbaiki masalah saat opsi Buat laporan tidak ditampilkan di UI.
Teradata	Amazon Redshift	Peningkatan konversi CAST fungsi.
Teradata	Amazon Redshift	Memperbaiki konversi yang rusak untuk <code>ST_Line_Interpolate_Point</code> .
Teradata	Amazon Redshift	Menghapus nilai tak terduga dari jalur pustaka Python.
Teradata	Pergeseran Merah Amazon RSQL	Memperbaiki kesalahan resolver yang muncul selama konversi beberapa FastLoad skrip.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Peningkatan konversi tipe data DATABASE perintah dan geometri.
Teradata BTEQ	AWS Glue	Memperbaiki masalah dengan sinkronisasi skrip sumber dan target yang salah di UI.

Masalah diselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 656

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan dukungan dari beberapa sumber dan basis data target dalam satu proyek. Pengguna sekarang dapat membuat aturan pemetaan untuk mencocokkan skema database yang berbeda dan platform target dalam proyek yang sama.
Semua	Semua	Menambahkan dukungan platform basis data target virtual. Pengguna sekarang tidak perlu terhubung ke database target untuk melihat bagaimana AWS SCT mengubah skema basis data sumber mereka.
Semua	Semua	Perbaikan UI: <ul style="list-style-type: none"> • Menambahkan Connect ke server dan Disconnect dari opsi server ke pohon metadata sumber dan target. • Menambahkan opsi untuk menghapus server database dari AWS SCT proyek.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Cassandra	Amazon DynamoDB	Menyelesaikan masalah pencarian di mana CASSANDRA_HOME variabel tidak menyertakan garis miring (/) setelah cassandra .yaml atau conf folder.
Cassandra	Amazon DynamoDB	Menambahkan dukungan Amazon Machine Image (AMI) untuk Amazon Linux 2.
Cassandra	Amazon DynamoDB	Pesan kesalahan yang ditingkatkan diberikan ketika kunci yang salah diberikan untuk Cassandra.
Cassandra	Amazon DynamoDB	Peningkatan konversi dengan mengubah properti dalam cassandra -env .yaml file tergantung pada versi database target.
Cassandra	Amazon DynamoDB	Meningkatkan versi Java pada target Cassandra Datacenter ke 1.8.0.
Greenplum	Amazon Redshift	Strategi pengoptimalan yang ditingkatkan di Pengaturan Proyek.
Greenplum	Amazon Redshift	Menyelesaikan masalah migrasi data di mana objek tidak diterapkan ke database dengan kesalahan ini: An I/O error occurred while sending to the backend .
Greenplum Microsoft SQL Server DW	Amazon Redshift	Menyelesaikan masalah di mana Apply RTRIM to string columns opsi tidak ditampilkan di UI.
SQLServer Microsoft	Babelfish untuk Aurora PostgreSQL	Menambahkan dukungan untuk Babelfish untuk Aurora SQL Postgre sebagai platform target. Pengguna sekarang dapat membuat laporan penilaian untuk memperkirakan migrasi dari SQL Server ke Babelfish untuk Aurora Postgre. SQL
Netezza	Amazon Redshift	Strategi pengoptimalan yang ditingkatkan di Pengaturan Proyek.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SAP ASE	Aurora PostgreSQL PostgreSQL	Menerapkan kemampuan untuk menghasilkan nama unik untuk indeks.
SAP ASE	Aurora PostgreSQL PostgreSQL	Memperbaiki masalah dengan kolom indeks duplikat dalam skrip target.
Kepingan salju	Amazon Redshift	Menyelesaikan masalah di mana Sembunyikan skema kosong, Sembunyikan database kosong, dan opsi Sembunyikan database sistem/skema tidak ditampilkan di UI.
Teradata	Pergeseran Merah Amazon RSQL	Menambahkan dukungan untuk konversi skrip MultiLoad pekerjaan Teradata ke skrip Amazon Redshift. RSQL
Teradata	Pergeseran Merah Amazon RSQL	Memperbaiki masalah dengan konversi variabel substitusi dalam FastLoad dan FastExport skrip.
Teradata	Pergeseran Merah Amazon RSQL	Memperbaiki masalah saat item tindakan tidak ditampilkan di tab Item Tindakan setelah beralih dari tab Ringkasan.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah saat terjadi kesalahan setelah menghasilkan laporan selama konversi FastExport skrip.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Pergeseran Merah Amazon RSQL	Mengatasi masalah pemformatan setelah konversi skrip shell.
Teradata	Pergeseran Merah Amazon RSQL	Memperbaiki masalah sehingga AI 13177 sekarang dikomentari dalam skrip yang dikonversi.
Teradata	Amazon Redshift	Memperbaiki konversi tabel temporal yang rusak.
Teradata	Amazon Redshift	Peningkatan konversi SET QUERY_BAND pernyataan.
Teradata	Amazon Redshift	Memperbaiki konversi NORMALIZE operasi yang rusak.
Vertica	Amazon Redshift	Memperbaiki deskripsi AI 17008.

Masalah diselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 655

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Pergeseran Merah Amazon RSQL	Memperbaiki masalah untuk memastikan semua masalah penilaian muncul dalam laporan saat FastLoad atau MultiLoad digunakan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Pergeseran Merah Amazon RSQL	Menambahkan dukungan untuk konversi skrip FastExport pekerjaan Teradata ke skrip Amazon Redshift. RSQL
Teradata	Pergeseran Merah Amazon RSQL	Memperbaiki masalah untuk memastikan tindakan Simpan manifes ke S3 diaktifkan dalam mode offline saat menggunakan FastLoad.
Teradata	Pergeseran Merah Amazon RSQL	Memperbaiki masalah untuk memastikan aturan pemetaan diterapkan untuk skrip seperti. FastLoad
Greenplum	Amazon Redshift	Meningkatkan versi driver minimum yang didukung untuk Greenplum menjadi 42.2.5.
Greenplum	Amazon Redshift	Menambahkan koneksi ke Greenplum via SSL dengan driver versi 42.2.5 atau lebih tinggi.
Oracle DW	Amazon Redshift	Peningkatan dukungan untuk mengeksekusi fungsi skalar yang ditentukan pengguna kustom () UDF di dalam yang lain. UDF
Oracle DW	Amazon Redshift	Memperbaiki masalah saat fungsi tidak diterapkan ke database dengan kesalahan ini: Failed to compile udf .
Oracle DW	Amazon Redshift	Peningkatan konversi dengan menggunakan deklarasi tipe yang sesuai seperti, pls -type untuk ROWTYPE parameter%.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah saat masalah penilaian tipe informasi tidak ditampilkan dalam laporan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Pergeseran Merah Amazon RSQL	Mengatasi kesalahan transformator setelah mengonversi beberapa skrip.
Teradata	Pergeseran Merah Amazon RSQL	Memperbaiki masalah sehingga masalah sekarang dikomentari dalam skrip yang dikonversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana FastExport EXPORT -> -> 'null' ditampilkan sebagai gantinya 'CAST' setelah konversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana beberapa fungsi paket ekstensi gagal saat diterapkan Cause: [JDBC Driver]String index out of range: 0 jika menggunakan driver versi 1.2.43
Teradata	Amazon Redshift	SETkonversi tabel- emulasi SET tabel ditambahkan untuk pernyataan insert-select.
Teradata	Amazon Redshift	CAST—mendukung pengecoran tipe data tambahan.
Teradata	Amazon Redshift	Memperbaiki konversi yang rusak untuk "other_current_time_01"
Teradata	Amazon Redshift	Teradata FastExport - Amazon RSQL Redshift: Peningkatan konversi perintah Teradata— FastExport FIELD
Teradata	Amazon Redshift	Teradata FastExport - Amazon RSQL Redshift: Peningkatan konversi perintah Teradata— FastExport LAYOUT

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Postgre SQL Aurora PostgreSQL	Menyelesaikan masalah di mana skrip target objek dengan SAVE EXCEPTIONS STATEMENT diubah setelah konversi ulang.
Oracle	Postgre SQL Aurora PostgreSQL	Menyelesaikan masalah di mana bidang yang salah ditentukan dalam klausa ORDER BY setelah proc_cursor_with_calc_columns konversi.
Oracle	Postgre SQL Aurora PostgreSQL	Terselesaikan: deklarasi aws_oracle_ext\$array_id\$temporary variabel tambahan diperlukan dalam ASSOCIATIVE COLLECTION konversi.
Oracle	Postgre SQL Aurora PostgreSQL	Terselesaikan: konversi yang salah dari a PRIMARY KEY dengan nama yang sama dari yang INDEX dimiliki oleh tabel yang sama.

Masalah diselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 654

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Postgre SQL Aurora PostgreSQL	Menyelesaikan masalah dengan Hierarchical Query PRIOR Pseudocolumns, kolom parsing error.
Oracle	Postgre SQL Aurora PostgreSQL	Menyelesaikan masalah untuk mengonversi komentar multi-baris dengan benar yang berisi garis miring dan tanda bintang (/ *).
Oracle	Postgre SQL Aurora PostgreSQL	Menambahkan tampilan sistem USER COL __ COMMENTS emulasi ke paket ekstensi.
Oracle	Postgre SQL Aurora PostgreSQL	Peningkatan konversi literal yang dikutip.
DB2 LUW	Postgre SQL Aurora PostgreSQL	Peningkatan konversi LABEL pernyataan yang menambah atau mengganti label dalam deskripsi tabel, tampilan, alias, atau kolom.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Tidak ada	Diganti. SYS USER\$ tabel sistem dengan USERS tampilan DBA _, dan kueri yang ditingkatkan.
Oracle DW	Amazon Redshift	Kueri metadata Oracle DW yang diperbarui.
Teradata	Pergeseran Merah Amazon RSQL	Menambahkan dukungan untuk konversi skrip shell, Teradata FastLoad, dan Teradata Basic Teradata Query () ke skrip Amazon Redshift. BTEQ RSQL
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Masalah teratasi di mana "merge_01" salah dikonversi.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Masalah teratasi sehingga End atau Identify (EOI) muncul di akhir skrip pada baris baru.
Sinaps Azure	Amazon Redshift	Pesan kesalahan yang ditingkatkan diberikan ketika kata sandi yang salah diberikan untuk Azure Synapse.
Teradata	Amazon Redshift	Peningkatan konversi UPDATE pernyataan untuk meneruskan nama alias yang tepat per standar Teradata.
Teradata	Amazon Redshift	Mengatasi kesalahan konversi kursor di mana tindakan tidak diterima.
Teradata	Amazon Redshift	Menyelesaikan masalah saat OVERLAP konversi NORMALIZE TD_ _ menjatuhkan baris.
Teradata	Amazon Redshift	Sekarang mendukung pemeriksaan tanggal yang ketat untuk DATE fungsi TO_ yang ditingkatkan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Peningkatan konversi fungsi Built-in TO_NUMBER (n).
Teradata	Amazon Redshift	Menyelesaikan masalah di mana kategori Skema tidak ada di pohon metadata.
Greenplum	Amazon Redshift	Menambahkan pilihan GP_SEGMENT_ID ke daftar saat membuat partisi virtual untuk tabel Greenplum.
Greenplum	Amazon Redshift	Menyelesaikan masalah di mana fungsi tidak diterapkan pada target.
MS SQL Server DW	Amazon Redshift	Menyelesaikan masalah di mana kesalahan transformasi terjadi setelah konversi tanpa AI 9996.
MS SQL Server DW	Amazon Redshift	Menyelesaikan masalah saat kesalahan dicatat saat membuka wizard paket ekstensi.
MS SQL Server DW	Amazon Redshift	Menyelesaikan masalah ketika gaya komentar yang salah digunakan untuk fungsi Redshift Python.
Netezza	Amazon Redshift	Menyelesaikan masalah di mana paket ekstensi Netezza—Redshift dengan profil gagal dibuat. AWS
Teradata	Pergeseran Merah Amazon RSQL	Peningkatan konversi FastLoad SESSIONS perintah.
Teradata	Pergeseran Merah Amazon RSQL	Laporan penilaian FastLoad skrip yang ditingkatkan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Pergeseran Merah Amazon RSQL	Diimplementasikan FastLoad WRITER Simpan ke tindakan S3.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana tombol FastLoad Simpan Skrip\ Simpan manifes ke s3 tidak aktif.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana FastLoad multifile_script hanya membuat satu file manifes setelah konversi, bukan tiga file yang diharapkan.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah di mana FastLoad folder tambahan ditampilkan di jalur S3.
Teradata	Pergeseran Merah Amazon RSQL	Menyelesaikan masalah yang FastLoad memiliki nama file manifes yang salah di jalur S3.

Masalah diselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 653

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	PostgreSQL	Menerapkan kemampuan untuk mengkonversi dinamis yang SQL dibuat dalam fungsi atau prosedur yang disebut.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
	Aurora Postgre SQL	
Oracle	Postgre SQL Aurora Postgre SQL	Peningkatan SQL konversi Dinamis: Dalam parameter sebagai variabel mengikat.
Oracle DW 18, 19	Amazon Redshift	Peningkatan konversi Oracle ke Redshift diterapkan: peningkatan konversi bawaan. AgregatLISTAGG; AnalitikLISTAGG.
Oracle DW 18,19	Amazon Redshift	Peningkatan konversi Oracle ke Redshift diterapkan: Kueri fitur baru.
Vertica	Amazon Redshift	Peningkatan konversi Vertica ke Redshift diterapkanSSL: JDBC untuk koneksi SSL dengan =true.
MS SQL Server DW	Amazon Redshift	Peningkatan konversi MS SQL Server ke Redshift: Tabel Eksternal.
Teradata	Amazon Redshift	Peningkatan konversi Teradata ke Redshift: operasi aritmatika tipe INTERVAL data.
Teradata	Amazon Redshift	Peningkatan konversi Teradata ke Redshift: Dukungan untuk alias kolom lateral.
Oracle	Tidak ada	Kueri Loader berikut sekarang digunakan DBA_USERS sebagai pengganti: SYS.USER\$ <ul style="list-style-type: none"> • get-tree-path-list- by-name-path .sql • estimate-table-or-view- constraints-by-schema .sql • estimate-table-or-view- constraints-by-selected-schemas .sql

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Peningkatan penyelarasan komentar saat SCT mengonversi makro Teradata ke prosedur tersimpan Redshift.
Oracle DW	Amazon Redshift	Peningkatan konversi elemen format Tanggal/Timestamp:., dan TO_DATE TO_TIMESTAMP TO_TIMESTAMP_TZ
Teradata	Amazon Redshift	Teratasi kesalahan konversi kursor Teradata.
Teradata	Amazon Redshift	Masalah teratasi yang menyebabkan atribut TD_NORMALIZE_OVERLAP dijatuhkan selama konversi.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana MAX fungsi diabaikan saat SCT mengonversi kueri.
Teradata	Amazon Redshift	SCTsekarang mengkonversi fungsi Teradata CHARACTERS ke fungsi Redshift. LENGTH
Teradata	Amazon Redshift	SCTsekarang mendukung konversi FORMAT ke TO_CHAR untuk format yang paling umum digunakan.
Semua	Semua	Peningkatan konversi rutinitas terenkripsi.

Masalah diselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 652

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	Postgre SQL	Menambahkan penguncian aplikasi untuk sp_getapplock dan sp_releaseapplock fungsi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Tidak ada	Amazon Redshift	Perbaiki Command Line Interface (CLI): menerapkan mode Script Command.
Oracle	PostgreSQL Aurora PostgreSQL	Menerapkan parameter rutin pengambilan sampel di dalam dinamisSQL.
Oracle	PostgreSQL Aurora PostgreSQL	Perbaiki konversi ke dinamis yang SQL dibuat dalam fungsi atau prosedur yang disebut.
SQLServer Microsoft Oracle DB2 LUW	Aurora PostgreSQL	Setiap fungsi lambda diterapkan dan dikonfigurasi melalui kebijakan hanya sekali, dan fungsi lambda umum digunakan kembali untuk semua sumber yang memungkinkan.
DB2 LUW	PostgreSQL	Masalah teratasi yang menyebabkan pesan kesalahan, “9996 - Keperahan kritis - Kesalahan transformator terjadi” saat menggunakan DB2 LUW sebagai sumber.
Teradata	Amazon Redshift	Support untuk ekspresi tabel rekursif dalam peluncuran Amazon Redshift yang akan datang.
Sinaps Azure	Amazon Redshift	Aturan optimasi skema yang diterapkan.
Teradata	Amazon Redshift	Mendukung konversi Zona Waktu dari makro Teradata ke prosedur tersimpan Redshift.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Support aritmatika pada nilai-nilai. PERIOD
Teradata	Amazon Redshift	Support konversi ekspresi tabel umum rekursif Teradata (). RECURSIVE CTE
Teradata	Amazon Redshift	Support case sensitive identifier melalui pengaturan pengguna, <code>enable_case_sensitive_identifier</code> . Jadi, "COLUMN_NAME" dan "Column_Name" menjadi nama kolom yang berbeda.
Teradata	Amazon Redshift	Mengatasi masalah tipe data desimal sehingga bidang Desimal dikonversi dengan presisi yang sama.
Teradata	Amazon Redshift	Mengatasi masalah dengan konversi aritmatika interval sehingga pengurangan aritmatika interval dikonversi dengan benar.
Teradata	Amazon Redshift	Peningkatan Teradata NUMBER untuk DATE menyetik casting.
Teradata	Amazon Redshift	Peningkatan Teradata DATE untuk menyetik casting NUMBER
Teradata BTEQ	Amazon Redshift	Peningkatan konversi tipe PERIOD data.
Teradata	Amazon Redshift	Masalah yang diselesaikan dengan memuat metadata untuk tabel dengan GEOMETRY kolom sehingga sekarang dimuat dari Teradata dengan benar.
Teradata	Amazon Redshift	Support konversi pernyataan gabungan saat mengonversi makro Teradata ke prosedur tersimpan Redshift.
Teradata	Amazon Redshift	Peningkatan konversi makro sederhana saat bermigrasi dari Teradata ke Redshift.
Teradata	Amazon Redshift	Memastikan konversi UPDATE pernyataan Teradata meneruskan nama alias yang tepat sesuai standar Teradata.

Masalah diselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 651

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	AWS SCT Laporan yang disempurnakan untuk memperbarui tautan ke item tindakan konversi yang direkomendasikan yang terdaftar.
SQLServer MS	Postgre SQL	Ditambahkan dukungan untuk konversi STR() fungsi.
SQLServer MS	Postgre SQL	Menambahkan dukungan untuk mengonversi EXOR operator bitwise (di ^ Microsoft SQL Server) ke Postgre SQL sebagai operator. #
Oracle	Postgre SQL	Menyelesaikan masalah di mana <code>aws_oracle_ext.UNI STR(null)</code> fungsi paket AWS SCT ekstensi tergantung NULL pada target PostgreSQL. AWS SCT sekarang menangani NULL.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Peningkatan konversi dilakukan untuk menyelesaikan masalah di mana konversi Amazon Redshift RSQL MERGE memberikan kesalahan transformasi.
Oracle DW	Amazon Redshift	Menerapkan built-in yang disempurnakan.
Oracle DW	Amazon Redshift	Menambahkan penyempurnaan berbasis fitur metadata, termasuk partisi Daftar Otomatis (_ _ _), Daftar Multi-kolom (TBLPARTLIST _ _ _ _ AUTO) dan Referensi Interval (TBL_ PART _ MULTI _ _ LIST). TBL PART RANGE INTVAL REF
none	Amazon Redshift	Peningkatan batas tabel partisi partisi fisik yang digunakan untuk UNION ALL konversi.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Perbaikan konversi dilakukan pada ruang lingkup laporan Penilaian.
Teradata	Amazon Redshift	Perbaikan konversi dilakukan untuk konversi Teradata MACRO yang kompleks.
Teradata	Amazon Redshift	Peningkatan konversi makro Teradata ke prosedur tersimpan Amazon Redshift sambil berkomentar tidak didukung. SQL
Teradata	Amazon Redshift	Menyelesaikan masalah saat konversi makro Teradata ke prosedur tersimpan Amazon Redshift menghasilkan referensi nama alias yang salah.
Teradata	Amazon Redshift	Peningkatan konversi pernyataan TeradataQUALIFY.
Teradata	Amazon Redshift	Peningkatan konversi untuk meneruskan komentar ke Amazon Redshift dan mempertahankan riwayat perubahan yang dilakukan pada tampilan.
Teradata	Amazon Redshift	Menyelesaikan masalah di mana RESET WHEN klausa tidak menghasilkan konversi yang benar.
Teradata BTEQ	Amazon Redshift	Peningkatan konversi BTEQ skrip yang berisi MERGE pernyataan.
Teradata	Amazon Redshift	Menambahkan fungsi bawaan untuk meningkatkan konversi bidang tipe PERIOD data.
SQLServer Microsoft	Amazon Redshift	Pemetaan tipe data transformasi yang disempurnakan untuk tipe TIME data.
Semua	Semua	Menambahkan akses ke publikasi awal manual AWS Schema Conversion Tool CLIReferensi dalam PDF format. Lihat AWS Schema Conversion Tool CLIReferensi .

Masalah diselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 650

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	<p>Penggunaan agen ekstraktor yang diperbarui dan ditingkatkan, termasuk:</p> <ul style="list-style-type: none"> • Konfigurasi untuk digunakan dengan penyimpanan bersama dan agen penyalinan khusus. • Mengekspor dan mengimpor tugas ekstraksi data dari satu proyek ke proyek lainnya. • Support untuk Azure SQL Data Warehouse (Azure Synapse) sebagai sumber. • Menggunakan partisi Netezza asli. <p>Untuk informasi selengkapnya, lihat Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool.</p>
Semua	Amazon RDS Postgre 13 SQL	AWS SCT sekarang mendukung Amazon RDS Postgre SQL 13 sebagai target.
SQLServer Microsoft	Aurora Postgre SQL	Peningkatan konversi hasil yang ditetapkan dari prosedur Microsoft SQL Server ke target Aurora SQL Postgre.
Oracle DW	Amazon Redshift	Menerapkan peningkatan konversi Oracle ke Amazon Redshift.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle DW	Amazon Redshift	Peningkatan yang diterapkan untuk mengonversi SQL pernyataan dinamis.
Oracle DW	Amazon Redshift	Menerapkan perbaikan SQL UDF konversi.
Oracle DW	Amazon Redshift	Pesan yang diklarifikasi yang AWS SCT tidak mendukung konversi. EXTERNAL TABLES
Oracle DW	Amazon Redshift	Fungsi konversi bawaan yang disempurnakan.
Teradata BTEQ	Pergeseran Merah Amazon RSQL	Peningkatan parameter substitusi penanganan di dalam BTEQ skrip saat menggunakan. AWS SCT GUI
Microsoft SQL Server DW SQLServer Microsoft Biru langit Sinaps Azure	Semua	Memutakhirkan versi JDBC driver minimum yang didukung untuk Microsoft SQL Server, Azure, Azure Synapse.

Masalah diselesaikan:

- Teradata: Peningkatan tambahan konversi makro [] RESOLVED
- Karakter khusus yang lolos di target menyebabkan SQL kesalahan dan pengerjaan ulang diperlukan untuk menempatkannya kembali [] RESOLVED
- Perbaikan umum

Catatan rilis untuk AWS SCT Build 649

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SQL Server DW	Amazon Redshift	MSSQL ke peningkatan konversi Amazon Redshift untuk mendukung tabel temporal.
Oracle DW	Amazon Redshift	<p>Peningkatan fungsi bawaan yang diterapkan, seperti:</p> <p>Fungsi konversi</p> <ul style="list-style-type: none"> • UNTUK_ _ BINARY DOUBLE • UNTUK_ _ BINARY FLOAT • UNTUK_ NUMBER • UNTUK_ DATE • UNTUK_ TIMESTAMP • KE_ _ TZ TIMESTAMP • UNTUK_ DSINTERVAL • UNTUK_ YMINTERVAL • VALIDATE_CONVERSION
Oracle DW	Amazon Redshift	<p>Peningkatan fungsi yang diterapkan untuk Perkiraan Pemrosesan Kueri, seperti:</p> <p>Fungsi agregat</p> <ul style="list-style-type: none"> • ANY_VALUE • APPROX_COUNT_DISTINCT

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		<ul style="list-style-type: none"> • APPROX_COUNT_DISTINCT_DETAIL • APPROX_COUNT_DISTINCT_AGG • LISTAGG • KE_ _ APPROX _ COUNT DISTINCT
Teradata	Amazon Redshift	Peningkatan konversi yang diterapkan untuk pemilihan kunci pengurutan dan distribusi otomatis Teradata. Mesin DB secara otomatis memilih distribusi dan mengurutkan kunci. Memperkenalkan tombol radio berlabel Gunakan penyetelan tabel otomatis Amazon Redshift ke pengaturan proyek saat ini > Strategi pengoptimalan > dialog Strategi Pemilihan Kunci Awal.
Teradata	Amazon Redshift	Pemuat AWS SCT tabel yang disempurnakan untuk memastikan AWS SCT memuat semua tabel dari Teradata.
Teradata	Amazon Redshift	Peningkatan konversi yang diterapkan sehingga Amazon Redshift mendukung pola subquery berkorelasi yang menyertakan klausa sederhana. WHERE NOT EXISTS
Teradata	Amazon Redshift	Menambahkan dukungan untuk penggunaan ECHO perintah di makro.
DB2 LUW	Postgre SQL Aurora Postgre SQL	<p>Dukungan yang diterapkan untuk DYNAMIC RESULTS SETS konversi, termasuk:</p> <ul style="list-style-type: none"> • Klausul cursor/TO WITH RETURN WITH RETURN CLIENT • DYNAMICRESULTSETSkonversi klausa rutin

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft Oracle DB2 LUW SAP ASE	Aurora Postgre SQL	Menerapkan dukungan untuk Aurora RDS SQL Postgre saat ini sebagai target.
SQLServer Microsoft Oracle DB2 LUW SAP ASE	MariaDB	Menerapkan dukungan untuk MariaDB 10.5 sebagai target.
SQLServer Microsoft	MariaDB	Diimplementasikan dukungan dari INSERT - RETURNING yang mengembalikan set hasil dari baris dimasukkan.
Oracle	Aurora Postgre SQL	Ditambahkan dukungan XMLFOREST fungsi untuk mengkonversi dari Oracle ke Aurora Postgre. SQL

Masalah diselesaikan:

- Perbaikan umum.

Catatan rilis untuk AWS SCT Build 648

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Postgre SQL	Mode penerapan kustom paket SQL ekstensi Aurora Postgre diterapkan: operator untuk tipe numerik/tanggal dan teks.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
	Amazon Aurora SQL Postgre -Edisi Kompatibel	
Oracle SQLServer Microsoft DB2 LUW	Aurora Postgre SQL	Konfigurasi Aurora Postgre SQL Lambda Invoke diimplementasikan: pembuatan ekstensi aws_lambda; penetapan peran ke cluster Aurora Postgre. IAM SQL <ul style="list-style-type: none"> • Oracle—Email, Pekerjaan,, Antrian, File WebAgent • DB2—Email, Tugas, File • Microsoft SQL Server- Email, Agen
Oracle	Postgre SQL	FORALLrefactoring konversi pernyataan diimplementasikan: <ul style="list-style-type: none"> • FORALLpernyataan • FORALL ... SAVE EXCEPTIONS • RETURNINGINTOdengan BULK COLLECT • SQL% BULK _ pengumpulan EXCEPTIONS sistem
Oracle DW 18, 19	Amazon Redshift	Peningkatan konversi Oracle ke Amazon Redshift diterapkan: peningkatan konversi bawaan. AgregatLISTAGG; AnalitikLISTAGG.
Oracle DW 18,19	Amazon Redshift	Peningkatan konversi Oracle ke Amazon Redshift diterapkan: Kueri fitur baru.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Vertica	Amazon Redshift	Peningkatan konversi Vertica ke Amazon Redshift diterapkan SSL: JDBC untuk koneksi SSL dengan <code>=true</code> .
Microsoft SQL Server DW	Amazon Redshift	Peningkatan konversi Microsoft SQL Server ke Redshift: Tabel Eksternal.
Teradata	Amazon Redshift	Peningkatan konversi Teradata ke Redshift: operasi aritmatika tipe INTERVAL data.
Teradata	Amazon Redshift	Peningkatan konversi Teradata ke Redshift: Dukungan untuk alias kolom lateral.

Masalah diselesaikan:

- Perbaikan umum

Catatan rilis untuk AWS SCT Build 647

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	SQLServer Microsoft	RDSsekarang mendukung fitur Database Mail.
SQLServer Microsoft	Saya SQL	<p>Menerapkan nama maksimum setiap jenis pengenalan - Panjang maksimum nama objek (misalnya, tabel, kendala, kolom) di SQL Server adalah 128 karakter. Panjang maksimum nama objek di My SQL adalah 64 karakter. Untuk menulis objek yang dikonversi ke SQL database Saya, Anda perlu mempersingkat nama mereka. Untuk mencegah nama duplikat setelah memotong, Anda perlu menambahkan "checksum" dari nama objek asli ke nama baru.</p> <p>Potong nama lebih dari 64 karakter sebagai berikut:</p> <pre>[first N chars]() + "" + [checksum]()</pre>

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		<p>[first N chars] = 64 - 1 - [length of checksum string]</p> <p>Sebagai contoh: example_of_a_test_schema_with_a_name_length_greater_than_64_characters ?? example_of_a_test_schema_with_a_name_length_greater_than_64_9703</p>
Oracle	SQLSaya/ Aurora Saya SQL	Diimplementasikan beban dan konversi komentar pada objek penyimpanan. Misalnya, pemrosesan komentar pada Tabel, dan pemrosesan komentar pada kolom Tabel/Tampilan.
Teradata	Amazon Redshift	Ditambahkan dukungan untuk konversi tipe TIME data.
Teradata	Amazon Redshift	Peningkatan konversi - TD_NORMALIZE_OVERLAP diimplementasikan.
Microsoft SQL Server DW	Amazon Redshift	Perbaiki konversi - SELECT dengan WITH klausa; tanpa SELECT FROM
Semua	Semua	AWS SCT Data Migration Service Assessor (DMSA) — Fitur baru ini memungkinkan Anda mengevaluasi beberapa server dan menerima laporan ringkasan yang menunjukkan arah target terbaik untuk lingkungan Anda.
Semua	Semua	AWS SCT Wizard - Perbandingan target sekarang menunjukkan perbedaan antara target dalam tampilan tabel tunggal.
Semua	Semua	UI Filter Pohon - Filter metadata yang didesain ulang menangani pola penyaringan yang lebih kompleks.
Semua	Semua	Laporan Penilaian - Bagian Peringatan yang Dirancang Ulang memberikan deskripsi yang lebih baik dan pemahaman yang lebih jelas tentang suatu masalah.

Masalah diselesaikan:

- Perbaikan umum
- Ekstraktor Data — Subtugas gagal dengan `ConcurrentModificationException` [RESOLVED].
- Microsoft SQL Server to My SQL - panjang pengenalan maks [RESOLVED].

Catatan rilis untuk AWS SCT Build 646

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Postgre SQL	Implementasi model format TM yang ditingkatkan.
Oracle	Postgre SQL	Implementasi masker format SP memberikan dukungan dasar untuk akhiran SP, hanya untuk bahasa Inggris.
Oracle	Postgre SQL	Penanganan nama objek panjang Oracle - AWS SCT sekarang menangani nama objek panjang Oracle sesuai dengan atribut panjang pengenalan maksimum target.
	Amazon Redshift	Pengkodean Amazon Redshift AZ64 dengan AWS SCT - Menambahkan pengkodean AZ64 kompresi untuk beberapa tipe data
Teradata	Amazon Redshift	Menambahkan dukungan untuk konversi transaksi Implisit.
Teradata	Amazon Redshift	Menambahkan dukungan untuk fungsi built-in geospasial Teradata: Metode <code>ST_LineString</code>
Greenplum	Amazon Redshift	Konversi urutan Greenplum - Menambahkan item berikutnya ke tab Properti: nilai min, nilai maks, kenaikan, siklus.
Greenplum	Amazon Redshift	Resolver - Menambahkan pemecahan tipe data "char".
Greenplum	Amazon Redshift	Panjang konversi karakter - Diperbarui konversi PL/PG untuk SQL jenis karakter.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Greenplum	Amazon Redshift	Menyelesaikan masalah dengan pemilihan kunci distribusi Greenplum di mana tabel memiliki DISTRIBUTION KEY tetapi AWS SCT tidak dapat mengenali dan mengambil tabel sebagai RANDOMLY DISTRIBUTED
Teradata	Amazon Redshift	Dukungan kursor Teradata - Ditambahkan dukungan untuk konversi kursor.
Teradata	Amazon Redshift	Kolom Identitas - Ditambahkan dukungan untuk konversi kolom Identitas.
Teradata	Amazon Redshift	INTERVAL tipe data - Menambahkan dukungan untuk konversi tipe INTERVAL data.

Masalah diselesaikan:

- Perbaikan umum
- Greenplum: Tidak dapat menjalankan konversi karena kesalahan di log [RESOLVED].
- MSSQL- PostgreSQL: Kesalahan transformator saat mengonversi LAG fungsi []. RESOLVED
- MSSQL— PostgreSQL: SCOPE _ IDENTITY []RESOLVED.
- AWS SCT tergantung di proyek DW [RESOLVED].
- Perlu aturan pemetaan untuk menghapus spasi tambahan pada nama kolom di AWS SCT [RESOLVED].

Catatan rilis untuk AWS SCT Build 645

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Berikan solusi untuk menyelesaikan tampilan Teradata yang tidak memenuhi syarat sepenuhnya (melihat nama atau objek yang tidak memenuhi syarat sepenuhnya dalam tampilan).

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Ditambahkan dukungan ASCII fungsi untuk menghitung node.
Teradata	Amazon Redshift	Saat AWS SCT melihat data multi-byte dalam Teradata yang CHAR didefinisikan sebagai CHAR(N), data tersebut dikonversi menjadi di Amazon VARCHAR(3*N) Redshift.
Teradata	Amazon Redshift	Berikan CAST konversi Teradata antara tanggal dan angka. <ul style="list-style-type: none"> • <code>SELECT Cast('2020-07-17' AS BIGINT)</code> • <code>SELECT Cast(20200630 - 19000000 AS DATE)</code>
Teradata	Amazon Redshift	Support konversi tipe PERIOD data Teradata menjadi dua kolom Amazon TIMESTAMP Redshift: <ul style="list-style-type: none"> • <code>PERIOD(TIMESTAMP)</code> • <code>PERIOD(TIMESTAMP WITH TIMEZONE)</code>
Teradata	Amazon Redshift	Support konversi RANK fungsi Teradata dengan RESET WHEN klausa.
Teradata	Amazon Redshift	Peningkatan dukungan konversi tipe data eksplisit, dan implisit CASTs pada ekspresi. CAST
Teradata	Amazon Redshift	Laporkan pola subquery berkorelasi yang tidak didukung. Untuk informasi selengkapnya, lihat Subkueri berkorelasi di Panduan Pengembang Database Amazon Redshift.
tidak satupun	Amazon Redshift	Peningkatan tabel membatasi dukungan untuk jenis RA3 node.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	Menambahkan dukungan untuk ekstraksi data geospasial Teradata. Untuk informasi selengkapnya, lihat Mengkueri data spasial di Amazon Redshift di Panduan Pengembang Database Amazon Redshift.
SQLServer Microsoft	Postgre SQL	Ditambahkan pilihan, <code>convert_procedures_to_function</code> .

Masalah diselesaikan:

- Perbaikan umum

Catatan rilis untuk AWS SCT Build 644

Perubahan untuk AWS SCT rilis 1.0.643 digabungkan menjadi rilis 1.0.644. AWS SCT

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	<p>Beberapa peningkatan konversi.</p> <ul style="list-style-type: none"> • Peningkatan konversi dengan QUALIFY dengan alias tabel. • Konversi yang ditingkatkan dengan IN operator. • Peningkatan konversi dengan LIKE operator. • Konversi yang ditingkatkan dengan menyoroti masalah dalam kode yang dikonversi. • Konversi yang ditingkatkan dengan urutan yang tidak biasa dari WHERE, QUALIFY klausa di. SQL • Kesalahan transformator tetap terjadi selama konversi JOIN() konstruksi prosedur <code>UPD_FT_SVC_TRANS_BH_CBH_IND</code> .

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		<ul style="list-style-type: none"> Peningkatan konversi makro ke prosedur yang disimpan. <p>Menambahkan AWS SCT CLI perintah khusus yang dapat mengurai skrip sql/bteq yang disediakan dan menghasilkan laporan tentang jumlah struktur sintaks yang ditemui dalam kode sumber.</p> <ul style="list-style-type: none"> Hitungan BTEQ perintah Hitungan dari HANDLERS Hitungan CAST kasus DMLDDLHitungan/kasus Hitung tampilan DMLs yang dapat diperbarui <p>Menambahkan item tindakan laporan penilaian: Kolom Teradata dengan format tanggal khusus tidak didukung di Amazon Redshift.</p>
Oracle	Postgre / Aurora Postgre SQL SQL	<p>Menambahkan fungsionalitas untuk menyimpan skrip instalasi paket ekstensi.</p> <p>Mengubah tingkat keparahan untuk AI 5334.</p> <p>Peningkatan kinerja menggunakan catatan sebagai variabel paketIMPLEMENTATION .</p> <p>Ditambahkan XMLAGG dukungan fungsi agregasi</p>
IBMDB2	Postgre / Aurora Postgre SQL SQL	Menambahkan beban dan konversi komentar pada implementasi objek penyimpanan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
MS SQL DW	Amazon Redshift	<p>Peningkatan konversi: Masalah yang diselesaikan dengan PATINDEX.</p> <p>Perbaiki UI:</p> <ul style="list-style-type: none"> • Simpan SQL untuk implementasi pohon sumber. • Menambahkan logika tambahan untuk pembuatan skrip untuk beberapa file.
Vertica	Amazon Redshift	Peningkatan UI: Simpan SQL untuk implementasi pohon sumber.

Masalah diselesaikan:

- Perbaiki umum untuk konversi antara Teradata dan Amazon Redshift
- Perbaiki bug umum dan peningkatan UI

Catatan rilis untuk AWS SCT Build 642

Perubahan untuk AWS Schema Conversion Tool rilis 1.0.642.

Note

AWS Schema Conversion Tool (AWS SCT) perubahan build 1.0.642 berlaku untuk Windows, Ubuntu, dan Fedora. Tidak ada build 1.0.642 untuk macOS.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft SSIS	AWS Glue	Menerapkan konversi ETL paket Microsoft SQL Server Integration Services (SSIS) ke AWS Glue. Untuk informasi selengkapnya, lihat Konversi ke SSIS dengan AWS Glue AWS SCT .
Oracle	MariaDB/SQL	Menerapkan bagian SQL PL/deklarasi dalam WITH klausa.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
	MODE =/Saya/ Amazon ORACLE SQL Aurora Saya SQL	
Oracle	Postgre / Aurora Postgre SQL SQL	Menambahkan dukungan untuk <code>DBMS_SESSION.RESET_PACKAGE</code> dan <code>DBMS_SESSION.MODIFY_PACKAGE</code> .
Vertica	Amazon Redshift	Aktifkan ekspor SQL skrip dari database Vertica ke Amazon Redshift.

Masalah diselesaikan:

- Peningkatan Laporan Penilaian.
- Peningkatan UI Laporan Penilaian.
- Tambahkan kemampuan untuk mengubah JVM pengaturan dari UI.
- Perbaikan umum.

Catatan rilis untuk AWS SCT build 641

Perubahan untuk AWS Schema Conversion Tool rilis 1.0.641.

Note

AWS Schema Conversion Tool (AWS SCT) perubahan build 1.0.641 berlaku untuk Windows, Ubuntu, dan Fedora. Tidak ada build 1.0.641 untuk macOS.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle/ms /Saya/Pos tgre/Db2 SQL SQL SQL LUW	Semua	Menghasilkan perhitungan Laporan Waktu dalam file.csv.
Teradata	Amazon Redshift	Ditambahkan dukungan untuk CSUM fungsi. Menambahkan dukungan untuk tipe data geospasial Teradata.
Teradata	Semua	Ditambahkan dukungan untuk mengkonversi IDENTITY kolom.
Greenplum	Amazon Redshift	Menambahkan dukungan untuk gaya distribusi AUTO selama konversi tabel Greenplum.
SAP ASE	Semua	Menghasilkan perhitungan Laporan Waktu dalam file.csv.

Terselesaikan:

- Berbagai perbaikan bug
- Berbagai peningkatan kinerja.

Catatan rilis untuk AWS SCT Build 640

Perubahan untuk AWS SCT rilis 1.0.633, 1.0.634, 1.0.635, 1.0.636, 1.0.637, 1.0.638, 1.0.639, dan 1.0.640 digabungkan menjadi rilis 1.0.640. AWS SCT

Note

AWS SCT perubahan build 1.0.640 berlaku untuk Windows, Ubuntu, dan Fedora. Mereka tidak berlaku untuk macOS.

Anda tidak dapat menginstal AWS SCT versi 1.0.640 atau lebih tinggi di Apple macOS. AWS SCT versi 1.0.632 adalah versi terakhir yang mendukung instalasi di Apple macOS.

Dalam tabel berikut, Anda dapat menemukan daftar fitur dan perbaikan bug untuk AWS Schema Conversion Tool versi yang telah digabungkan menjadi rilis 1.0.640. Tabel ini mengelompokkan fitur dan perbaikan bug oleh mesin sumber.

Topik

- [Rilis 1.0.640 perubahan Oracle](#)
- [Rilis 1.0.640 perubahan Microsoft Server SQL](#)
- [Rilis 1.0.640 Perubahan Saya SQL](#)
- [Rilis 1.0.640 Perubahan Postgre SQL](#)
- [Rilis 1.0.640 perubahan Db2 LUW](#)
- [Rilis 1.0.640 Perubahan Teradata](#)
- [Rilis 1.0.640 perubahan untuk mesin lain](#)

Rilis 1.0.640 perubahan Oracle

Daftar tabel berikut membangun 1.0.640 perubahan di mana Oracle adalah mesin sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Postgre SQL Aurora Postgre SQL	Konversi SQL kode yang diterapkan dalam aplikasi Java dan Pro* C.
Oracle	Postgre SQL Aurora Postgre SQL	Peningkatan kinerja fungsi berikut bila digunakan dalam WHERE klausa: <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>

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	RDS MariaDB 10.4	Menambahkan dukungan RDS MariaDB 10.4 untuk semua vendor pemrosesan transaksional online (). OLTP
Oracle	Postgre / Aurora Postgre SQL SQL	Menambahkan dukungan untuk DBMS _UTILITY. GET_TIME. Menambahkan emulasi berikut: <ul style="list-style-type: none"> • DBMS _UTILITY.GET_TIME • DBMS _UTILITY.FORMAT_CALL_STACK • DBMS _UTILITY.CURRENT_INSTANCE
Oracle	MariaDB/S aya/Aurora My SQL / Microsoft Server SQL Mode=Oracle/Postgre/Aurora SQL Postgre/Oracle SQL SQL RDS	Menambahkan dukungan klausa berbagi untuk TABLE (DATA, EXTENDED DATA), VIEW (DATA, EXTENDED DATA), dan SEQUENCE (DATA)
Oracle	Postgre SQL / Aurora Postgre / Oracle SQL RDS	DEFAULT Definisi kolom dapat diperluas agar diterapkan untuk NULL penyisipan eksplisit. DEFAULT DEFAULT Klausul ini memiliki klausa ON NULL baru. Klausul baru ini menginstruksikan database untuk menetapkan nilai kolom default tertentu ketika INSERT pernyataan mencoba untuk menetapkan nilai yang dievaluasi. NULL

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	MariaDB/ MariaDB (=) SQL MODE ORACLE	Ditambahkan dukungan untuk “Kolom Identitas,” yang secara otomatis meningkat pada saat penyisipan.
Semua	Semua	Tingkatkan ke Amazon JDK Corretto 11 dari 8. JDK Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
Oracle	MariaDB 10.2/Mari aDB 10.3/ Saya/ Aurora SQL Saya/ Postgre / Aurora Postgre SQL SQL SQL	DEFAULTKlausula memiliki ON NULL klausula baru, yang menginstruksikan database untuk menetapkan nilai kolom default tertentu ketika INSERT pernyataan mencoba untuk menetapkan nilai yang dievaluasi. NULL
Oracle	Oracle RDS / Saya/Aur ora SQL Saya/Post gre/Auror a Postgre SQL SQL SQL	Ditambahkan dukungan untuk IDENTITY kolom.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	SQL8.x saya	Ditambahkan dukungan untuk CHECK kendala.
Oracle	Postgre / Aurora Postgre SQL SQL	<p>Diimplementasikan memeriksa ANYDATA NULL IS/IS NOT NULL menggunakan paket ekstensi rutin.</p> <p>Menerapkan emulasi VALUE fungsi yang digunakan dalam query berdasarkan TABLE fungsi. XMLSequence</p> <p>Ditambahkan DBMS _ LOB dukungan untuk rutinitas bawaan berikut:</p> <ul style="list-style-type: none"> • DBMS_LOB.CREATETEMPORARY • DBMS_LOB.FREETEMPORARY • DBMS_LOB.APPEND
Semua	SQLPelade n	<p>SQLServer 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE _ FOR _ SEQUENTIAL _ KEY.</p> <p>SQLServer 2017: Menambahkan dukungan untuk tipe tabel Graph Databases Node dan Edge.</p> <p>SQLServer 2016: Menambahkan dukungan untuk TEMPORAL TABLES.</p>
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Amazon Redshift	<p>Menerapkan konversi atribut kursor di blok bersarang.</p> <p>Amazon Redshift tidak mendukung koleksi. Variabel terkait dikonversi sebagai VARCHAR. Semua operasi koleksi selain menetapkan satu variabel ke variabel lain ditolak, termasuk inisiasi dan akses elemen koleksi.</p> <p>Menerapkan gaya distribusi Amazon Redshift =. AUTO</p>

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Postgre / Aurora Postgre SQL SQL	<p>Jika kata nonreserved di Oracle dicadangkan di PostgreSQL, maka berikut ini benar:</p> <ul style="list-style-type: none"> • Jika kata tersebut dikutip, ia mempertahankan kasusnya dan tetap dikutip. • Jika kata itu tidak dikutip, itu dilemparkan ke huruf besar dan dikutip. <p>Menerapkan kemampuan untuk menggunakan fungsi sebagai input keLTRIM,RTRIM, dan TRIM fungsi.</p> <p>SELECTDISTINCT, Ekspresi ORDER BY harus muncul di daftar pilih.</p> <p>Untuk parameter kursor yang mengikuti setelah parameter dengan DEFAULT nilai, AWS SCT tambahkan klausa DEFAULT IS NULL</p> <p>Parameter OUT kursor sumber dikonversi ke parameter kursor IN.</p> <p>Variabel paket yang diimplementasikan kembali dengan menambahkan opsi "Package variable logic implementation" di bawah "Conversion settings". Pengaturan yang tersedia adalah: "variabel sesi" dan "objek global plv8". Defaultnya adalah "variabel sesi".</p> <p>Diimplementasikan AUTONOMOUS _ dukungan TRANSACTION pragma dengan dblink dan pg_background.</p>
Oracle	Semua	Tampilan diimplementasikan SYS_%_TAB. COMMENTS
Oracle	Postgre SQL	Input variabel ke filter tidak didukung di SQL Postgre. Saat mengonversi dari Oracle ke PostgreSQL, jika filter variabel ditemui, pengecualian sekarang dilaporkan.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Oracle	Amazon Redshift	<p>Kode tersimpan yang diimplementasikan FOR.. LOOP Peningkatan konversi kursor.</p> <p>Diimplementasikan pemanggilan kode tersimpan fungsi/prosedur dengan parameter default.</p> <p>Menerapkan kemampuan kode tersimpan UPDATE dengan alias tanpa WHERE klausa.</p> <p>Fungsi kode tersimpan yang diimplementasikan membentuk kasus tambahan dengan SELECT FROM ganda.</p> <p>Diimplementasikan kode tersimpan Tabel% ROWTYPE parameter dan variabel paket.</p> <p>Menerapkan kode tersimpan yang digunakan JAVA dan prosedur eksternal.</p> <p>Diimplementasikan paket Oracle standar dalam kode yang disimpan.</p>

Rilis 1.0.640 perubahan Microsoft Server SQL

Daftar tabel berikut membangun 1.0.640 perubahan di mana Microsoft SQL Server adalah mesin sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Microsoft Azure/ Server	Postgre SQL / Aurora	Ditambahkan dukungan untuk COLUMN STORE indeks.
Microsoft SQL	Postgre / Saya/Aurora Saya SQL SQL SQL	

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLServer Microsoft	RDSMariaD B 10.4	Menambahkan dukungan RDS MariaDB 10.4 untuk semua vendor pemrosesan transaksional online (). OLTP
Azure/Ser ver SQL	SQLMariaD b/saya/Au rora Saya/ Postgre/ Aurora Postgre SQL SQL SQL	Menambahkan dukungan untuk atribut KEY indeks OPTIMIZE FOR SEQUENTIAL _ _ _ _.
Azure/Ser ver SQL	SQLSaya/ Aurora Saya/Post gre/Auror a SQL Postgre SQL SQL	Menambahkan dukungan untuk Database Node dan tipe tabel Edge.
Azure/Ser ver SQL	SQLMariaD b/saya/Au rora Saya/ Postgre/ Aurora Postgre SQL SQL SQL	Menambahkan dukungan untuk TEMPORAL TABLES.
Semua	Semua	Tingkatkan ke Amazon JDK Corretto 11 dari 8. JDK Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
Azure/Server SQL	SQLSaya/Aurora Saya/Postgre /Aurora SQL Postgre SQL / MariaDB SQL	Ditambahkan dukungan untuk DML pengolahan untuk SQL Server Graph Architecture.
SQLPeladen	Aurora Postgre SQL	Ditambahkan pilihan untuk mengkonversi parameter tanpa par_ awalan.
Azure/Server SQL	SQL8.x saya	Ditambahkan dukungan untuk CHECK kendala.
Semua	SQLPeladen	<p>SQLServer 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE _ FOR _ SEQUENTIAL _ KEY.</p> <p>SQLServer 2017: Menambahkan dukungan untuk tipe tabel Graph Databases Node dan Edge.</p> <p>SQLServer 2016: Menambahkan dukungan untuk TEMPORAL TABLES.</p>
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLPelade n	AWS Glue (Cangkang Python)	Perbaiki konversi, termasuk: <ul style="list-style-type: none"> • Menerapkan konversi fungsi bawaan ke Python.String. • Diimplementasikan EXECUTE dan EXEC dalam kode yang disimpan. • Diimplementasikan menggunakan jenis tabel.
Azure/Ser ver SQL	Postgre / Aurora Postgre SQL SQL	Diimplementasikan membuat TMP prosedur \$ opsional.
SQLPelade n	SQLSaya/ Aurora Saya SQL	Operasi aritmatika yang diperpanjang dengan tanggal. Emulasi konstruksi 'TOP(ekspresi). WITH TIES Setelah memanggil prosedur dengan refkursor yang dihasilkan keluar, refkursor sekarang ditutup. Menyetel tingkat GLOBAL isolasi tidak didukung di Aurora My. SQL Hanya lingkup sesi yang dapat diubah. Perilaku default transaksi adalah menggunakan REPEATABLE READ dan membaca secara konsisten. Aplikasi yang dirancang untuk digunakan READ COMMITTED mungkin perlu dimodifikasi. Atau, mereka dapat secara eksplisit mengubah default ke. READ COMMITTED

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
SQLPelade n	AWS Glue (Cangkang Python)	<p>SQLPernyataan server menghasilkan set hasil yang lengkap, tetapi ada kalanya hasilnya paling baik diproses satu baris pada satu waktu. Membuka kursor pada set hasil memungkinkan pemrosesan hasil yang ditetapkan satu baris pada satu waktu. Anda dapat menetapkan kursor ke variabel atau parameter dengan tipe data kursor.</p> <p>Diimplementasikan melampirkan serangkaian SQL pernyataan Transact- untuk kode yang disimpan sehingga sekelompok SQL pernyataan Transact- dapat dijalankan meskipun Python tidak mendukung SQL Server dan as. BEGIN END control-of-flow</p> <p>SQLServer LABEL dan GOTO pernyataan tidak didukung oleh AWS Glue. Jika AWS SCT menemukan label dalam kode, itu dilewati. Jika AWS SCT menemukan GOTO pernyataan, itu dikomentari.</p>
SQLPelade n	Amazon Redshift	<p>Menerapkan pemrosesan kondisional dari SQL pernyataan Transact- untuk kode yang disimpan dengan mengimplementasikan IF... ELSEkontrol.</p> <p>Diimplementasikan melampirkan serangkaian Transact- SQL pernyataan untuk kode yang disimpan sehingga sekelompok SQL pernyataan Transact- dapat dijalankan sebagai blok. Mendukung bersarangBEGIN... ENDblok.</p> <p>Diimplementasikan SET dan SELECT dalam kode yang disimpan.</p> <p>Diimplementasikan CREATE INDEX di Amazon Redshift (yang tidak mendukung indeks) dengan membuat kunci pengurutan yang ditentukan pengguna pada tabel.</p>

Rilis 1.0.640 Perubahan Saya SQL

Daftar tabel berikut membangun 1.0.640 perubahan di mana My SQL adalah mesin sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Saya SQL	Postgre 12.x SQL	Ditambahkan dukungan untuk kolom yang dihasilkan.
Semua	Semua	Tingkatkan ke Amazon JDK Corretto 11 dari 8. JDK Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
Saya SQL	SQLPostgre/Aurora Postgre 11. SQL	Ditambahkan dukungan untuk yang berikut: <ul style="list-style-type: none"> • Transaksi tertanam di dalam prosedur SQL tersimpan. • Kemampuan untuk CALL SQL menyimpan prosedur. • Kemampuan untuk membuat prosedur SQL tersimpan.
Semua	SQLPeladen	<p>SQLServer 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE _ FOR _ SEQUENTIAL _KEY.</p> <p>SQLServer 2017: Menambahkan dukungan untuk tipe tabel Graph Databases Node dan Edge.</p> <p>SQLServer 2016: Menambahkan dukungan untuk TEMPORAL TABLES.</p>
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.

Rilis 1.0.640 Perubahan Postgre SQL

Daftar tabel berikut membangun 1.0.640 perubahan di mana Postgre SQL adalah mesin sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Postgre SQL	SQL8.x saya	<p>SQLSekarang saya mendukung pembuatan bagian kunci indeks fungsional yang mengindeks nilai ekspresi daripada nilai kolom. Bagian kunci fungsional memungkinkan pengindeksan nilai, seperti JSON nilai, yang tidak dapat diindeks sebaliknya.</p> <p>Saya SQL sekarang mendukung Sekarang CTE dan RekursifCTE.</p>
Semua	Semua	Tingkatkan ke Amazon JDK Corretto 11 dari 8. JDK Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
Postgre 11.x SQL	SQLPostgre/Aurora Postgre 11. SQL	<p>Ditambahkan dukungan untuk yang berikut:</p> <ul style="list-style-type: none"> • Transaksi tertanam di dalam prosedur SQL tersimpan. • Kemampuan untuk CALL SQL menyimpan prosedur. • Kemampuan untuk membuat prosedur SQL tersimpan.
Postgre SQL	SQL8.x saya	<p>Menambahkan SQL dukungan saya untuk indeks turun. DESCdalam definisi indeks tidak lagi diabaikan, tetapi menyebabkan penyimpanan nilai-nilai kunci dalam urutan menurun.</p> <p>Menambahkan SQL dukungan Saya penggunaan ekspresi sebagai nilai default dalam spesifikasi tipe data, termasuk ekspresi sebagai nilai default untukBLOB,TEXT,GEOMETRY, dan tipe JSON data.</p> <p>Beberapa fungsi agregat yang ada sekarang dapat digunakan sebagai fungsi jendela:</p> <ul style="list-style-type: none"> • AVG() • BIT_AND() • BIT_OR()

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		<ul style="list-style-type: none"> • BIT_XOR() • COUNT() • JSON_ARRAYAGG() • JSON_OBJECTAGG() • MAX() • MIN() • STDDEV_POP() • STDDEV() • STD() • STDDEV_SAMP() • SUM() • VAR_POP() • VARIANCE() • VAR_SAMP() <p>Saya SQL mendukung fungsi jendela yang, untuk setiap baris dari kueri, melakukan perhitungan menggunakan baris yang terkait dengan baris itu.</p> <ul style="list-style-type: none"> • CUME_DIST() • DENSE_RANK() • FIRST_VALUE() • LAG() • LAST_VALUE() • LEAD() • NTH_VALUE() • NTILE() • PERCENT_RANK() • RANK()

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
		<ul style="list-style-type: none"> • ROW_NUMBER()
Postgre SQL	SQL8.x saya	Ditambahkan dukungan untuk CHECK kendala.
Semua	SQLPelade n	<p>SQLServer 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE _ FOR _ SEQUENTIAL _ KEY.</p> <p>SQLServer 2017: Menambahkan dukungan untuk tipe tabel Graph Databases Node dan Edge.</p> <p>SQLServer 2016: Menambahkan dukungan untuk TEMPORAL TABLES.</p>
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.
Postgre / Aurora Postgre SQL SQL	Semua	<p>Ditambahkan tampilan sistem sysindex emulasi.</p> <p>Jika ada SELECT pernyataan dalam prosedur tanpa menentukan INTO, parameter INOUT p_refcur tipe refcursor dibuat untuk prosedur pada target.</p>

Rilis 1.0.640 perubahan Db2 LUW

Tabel berikut mencantumkan perubahan build 1.0.640 yang DB2 LUW merupakan mesin sumber.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
DB2 LUW	RDSMariaDB 10.4	Menambahkan dukungan RDS MariaDB 10.4 untuk semua vendor pemrosesan transaksional online (). OLTP

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Tingkatkan ke Amazon JDK Corretto 11 dari 8. JDK Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
DB2 LUW	SQL8.0.17 saya	Menambahkan CHECK dukungan kendala.
Semua	SQLPelade n	<p>SQLServer 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE _ FOR _ SEQUENTIAL _KEY.</p> <p>SQLServer 2017: Menambahkan dukungan untuk tipe tabel Graph Databases Node dan Edge.</p> <p>SQLServer 2016: Menambahkan dukungan untuk TEMPORAL TABLES.</p>
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.

Rilis 1.0.640 Perubahan Teradata

Tabel berikut mencantumkan build 1.0.640 perubahan mesin sumber Teradata.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Teradata	Amazon Redshift	<p>Ditambahkan dukungan untuk MERGE dan QUALIFY pernyataan.</p> <p>Dihapus LOCKING ROWS FOR ACCESS klausa dari pernyataan Teradata.</p> <p>Ditambahkan dukungan untuk CAST fungsi.</p>

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Semua	Semua	Tingkatkan ke Amazon JDK Corretto 11 dari 8. JDK Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Teradata	Teradata	Peningkatan yang diterapkan di REGEXP _ INSTR () dan REGEXP _ SUBSTR ().
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
Semua	SQLPelade n	<p>SQLServer 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE _ FOR _ SEQUENTIAL _ KEY.</p> <p>SQLServer 2017: Menambahkan dukungan untuk tipe tabel Graph Databases Node dan Edge.</p> <p>SQLServer 2016: Menambahkan dukungan untuk TEMPORAL TABLES.</p>
Teradata	Semua	Menambahkan dukungan untuk REGEXP _ INSTR () dan REGEXP _ SUBSTR ().
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.
Teradata	Amazon Redshift	<p>Menerapkan kemampuan untuk menyimpan SQL pohon sumber ke dalam satu file atau beberapa file secara bertahap menggunakan pengaturan di Pengaturan Proyek, Simpan sebagai SQL dan Terapkan, Daftar dropdown: File tunggal/Beberapa file.</p> <p>Peningkatan yang diterapkan dalam konversi tampilan dan prosedur.</p>
Teradata	Semua	Ditambahkan dukungan untuk Teradata versi 16.20

Rilis 1.0.640 perubahan untuk mesin lain

Tabel berikut mencantumkan perubahan build 1.0.640 untuk mesin sumber lainnya.

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Sybase	RDSMariaDB 10.4	Menambahkan dukungan RDS MariaDB 10.4 untuk semua vendor pemrosesan transaksional online (). OLTP
SAP ASE	MariaDB	<p>Diimplementasikan sebagai berikut:</p> <ul style="list-style-type: none"> • MariaDB 10.4 • EXECUTEIMMEDIATEpernyataan • DEFAULTdefinisi • CHECKdukungan kendala
SAP ASE	PostgreSQL 12.x	Ditambahkan dukungan untuk kolom yang dihasilkan.
Semua	Semua	Tingkatkan ke Amazon JDK Corretto 11 dari 8. JDK Untuk informasi selengkapnya, termasuk tautan unduhan, lihat Apa itu Amazon Corretto 11? di Panduan Pengguna Amazon Corretto 11.
Semua	Semua	Menambahkan informasi ke laporan penilaian tentang kemungkinan inkonsistensi dalam database pengguna.
SAP ASE	SQL8.0.17	Menambahkan CHECK dukungan kendala.
Semua	SQLPeladen	<p>SQLServer 2019: Menambahkan dukungan untuk atribut indeks baru OPTIMIZE _ FOR _ SEQUENTIAL _ KEY.</p> <p>SQLServer 2017: Menambahkan dukungan untuk tipe tabel Graph Databases Node dan Edge.</p> <p>SQLServer 2016: Menambahkan dukungan untuk TEMPORAL TABLES.</p>

Sumber	Target	Apa yang baru, ditingkatkan, atau diperbaiki
Vertica	Amazon Redshift	Ditambahkan dukungan untuk gaya distribusi =AUTO.
Semua	Semua	Menerapkan kemampuan untuk mengganti partisi fisik dengan partisi virtual. Ekstraktor gudang data mengekstrak data sesuai dengan partisi virtual yang dibuat.
Amazon Redshift	Amazon Redshift	Fungsi bawaan yang tidak didukung dalam DML pernyataan diganti dengan NULL sebagai placeholder.
Sybase	Postgre SQL	Menambahkan dukungan untuk fungsi asli.
SAP ASE	SQLSaya/ Aurora Saya SQL	Tingkat isolasi default untuk Aurora My SQL adalah. REPEATABLE READ Menyetel tingkat GLOBAL isolasi tidak didukung di Aurora My. SQL Hanya lingkup sesi yang dapat diubah. Perilaku default transaksi adalah menggunakan REPEATABLE READ dan membaca secara konsisten. Aplikasi yang dirancang untuk dijalankan READ COMMITTED mungkin perlu dimodifikasi. Atau Anda dapat secara eksplisit mengubah default ke. READ COMMITTED
SAP ASE	Postgre SQL	Menambahkan dukungan untuk CONVERT fungsi (optimis) tanpa paket ekstensi.
SAP ASE	Semua	Ditambahkan tampilan sistem sysindex emulasi. Jika ada SELECT pernyataan dalam prosedur tanpa menentukan INTO, parameter INOUT p_refcur tipe refcursor dibuat untuk prosedur pada target.
Greenplum	Amazon Redshift	Diimplementasikan CREATE TEMPORARY TABLE sebagai berikut:

Fitur Baru di AWS Schema Conversion Tool

Tabel berikut menjelaskan perubahan penting pada AWS Schema Conversion Tool (AWS SCT) panduan pengguna setelah Januari 2018.

Anda dapat berlangganan RSS feed untuk diberitahu tentang pembaruan dokumentasi ini.

Perubahan	Deskripsi	Tanggal
AWS SCT membangun #1 .0.672	Build 1.0.672 menyediakan dukungan Amazon RDS untuk Postgre SQL 15 sebagai target dan Microsoft SQL Server versi 2022 sebagai sumber. Ini juga menambahkan dukungan untuk fitur Amazon Redshift baru dalam kode yang dikonversi, mengimple mentasikan beberapa peningkatan konversi untuk IBM Db2 untuk sumber z/OS, dan menyelesaikan sejumlah masalah konversi.	8 Mei 2023
AWS SCT membangun #1 .0.671	Build 1.0.671 menyediak an dukungan migrasi dari Apache Oozie ke. AWS Step Functions Ini juga menambahkan dukungan untuk BigQuery sebagai sumber untuk proses penilaian multiserver. Selain itu, ia menambahkan pengaturan konversi baru untuk IBM Db2 untuk z/OS sebagai sumber	8 Maret 2023

dan menyelesaikan sejumlah masalah konversi.

[AWS SCT membangun #1 .0.670](#)

Build 1.0.670 menyediakan dukungan migrasi dari Hadoop ke Amazon. EMR. Ini juga menambahkan dukungan untuk Azure Synapse Analytics sebagai sumber untuk proses penilaian multiserver. Selain itu, meningkatkan konversi SQL kode yang tertanam dalam aplikasi Java dan menyelesaikan sejumlah masalah konversi.

23 Januari 2023

[AWS SCT membangun #1 .0.669](#)

Build 1.0.669 mengimplementasikan dukungan partisi native untuk migrasi data dari gudang data Oracle. Ini juga meningkatkan proses penilaian multiserver, menambahkan fitur baru dalam agen ekstraksi data, dan menyelesaikan sejumlah masalah konversi.

19 Desember 2022

[AWS SCT membangun
#1 .0.668](#)

Build 1.0.668 mengimple-
mentasikan partisi virtual
otomatis untuk migrasi data
dari database Greenplum,
dan menambahkan dukungan
migrasi data dari database
Snowflake ke Amazon
Redshift. Ini juga meningkat-
kan konversi SQL kode yang
tertanam dalam aplikasi C #
dan menyelesaikan sejumlah
masalah konversi.

16 November 2022

[AWS SCT membangun
#1 .0.667](#)

Build 1.0.667 menyediak-
an dukungan untuk mesin
ekstrak, transformasi, dan
beban (ETL) Informatica
sebagai sumber migrasi. Ini
juga memperbarui versi paket
ekstensi, meningkatkan versi
driver minimum yang didukung
untuk Amazon Redshift, dan
menyelesaikan sejumlah
masalah konversi.

13 Oktober 2022

[AWS SCT membangun
#1 .0.666](#)

Build 1.0.666 meningkatkan
konversi aplikasi Java dengan
menambahkan dukungan
untuk kerangka kerja. MyBatis
Ini juga menambahkan
fungsi baru ke dalam paket
ekstensi, meningkatkan
pemuat metadata sumber,
dan menyelesaikan sejumlah
masalah konversi.

September 20, 2022

[AWS SCT membangun
#1 .0.665](#)

Build 1.0.665 menyediakan dukungan BigQuery sebagai sumber migrasi. Ini juga mengimplementasikan dukungan versi baru dari file konfigurasi fitur Babelfish. Selain itu, ini meningkatkan konversi gudang data ke Amazon Redshift, dan menyelesaikan sejumlah masalah konversi.

Agustus 29, 2022

[AWS SCT membangun
#1 .0.664](#)

Build 1.0.664 menyediakan dukungan Amazon Redshift Serverless sebagai sumber migrasi atau target. Ini juga mengimplementasikan penyeimbangan memori otomatis dalam tugas ekstraksi data, dan memperbaiki kesalahan di mana AWS SCT tidak dapat terhubung ke perangkat. AWS Snowball Selain itu, ia menambahkan kemampuan untuk mengubah pemeriksaan kolom dalam aturan migrasi, meningkatkan antarmuka pengguna, dan menyelesaikan sejumlah masalah konversi.

14 Juli 2022

[AWS SCT membangun
#1 .0.663](#)

Build 1.0.663 menambahkan dukungan untuk Babelfish untuk Aurora SQL PostgreSQL 1.2.0 dan meningkatkan kemampuan laporan penilaian multiserver. Ini juga menambahkan fitur baru dalam aturan migrasi, memperbaiki dua kesalahan pemuat, dan menyelesaikan sejumlah masalah konversi.

Juni 20, 2022

[AWS SCT membangun
#1 .0.662](#)

Build 1.0.662 mengimplementasikan konversi SQL kode dalam aplikasi C# dan meningkatkan alur kerja laporan penilaian multiserver. Ini juga menambahkan beberapa peningkatan konversi dan menyelesaikan sejumlah masalah konversi.

Mei 19, 2022

[AWS SCT membangun
#1 .0.661](#)

Build 1.0.661 menyediakan dukungan IBM Db2 untuk z/OS sebagai sumber migrasi. Ini juga menambahkan dukungan untuk konversi skrip ekstrak, transformasi, dan load (ETL) ke AWS Glue Studio dan menyelesaikan sejumlah masalah konversi.

21 April 2022

[AWS SCT membangun
#1 .0.660](#)

Build 1.0.660 memberikan dukungan Postgre mayor SQL versi 14 dan MariaDB 10.6 sebagai target migrasi. Ini juga menambahkan dukungan untuk konversi indeks Oracle untuk tampilan terwujud, dan menyelesaikan sejumlah masalah konversi.

Maret 21, 2022

[AWS SCT membangun
#1 .0.659](#)

Build 1.0.659 memberikan dukungan Postgre mayor SQL versi 13 pada Aurora SQL Postgre -Compatible Edition sebagai target migrasi. Ini mengimplementasikan konversi SQL kode dalam aplikasi C #, menambahkan dukungan Oracle Unified Audit, dan menyelesaikan sejumlah masalah konversi.

Februari 21, 2022

[AWS SCT membangun
#1 .0.658](#)

Build 1.0.658 menyediakan integrasi dengan AWS Secrets Manager dan menambahkan dukungan platform basis data target virtual Amazon Redshift. Ini juga menambahkan sejumlah perbaikan konversi dan perbaikan bug.

20 Januari 2022

[AWS SCT membangun
#1 .0.657](#)

Build 1.0.657 meningkatkan konversi dari Microsoft Server SQL ke Aurora PostgreSQL -Compatible Edition, RDS Amazon untuk PostgreSQL, dan tujuan migrasi lainnya. SQL Ini juga menambahkan sejumlah perbaikan antarmuka pengguna dan perbaikan bug.

Desember 20, 2021

[AWS SCT membangun
#1 .0.656](#)

Build 1.0.656 menyediakan dukungan dari beberapa sumber dan basis data target dalam satu proyek. Ini juga menambahkan konversi, strategi pengoptimalan, dan perbaikan umum dan sejumlah perbaikan bug.

22 November 2021

[AWS SCT membangun
#1 .0.655](#)

Build 1.0.655 mengimplementasikan konversi skrip pekerjaan Teradata FastExport ke Amazon Redshift RSQL dan meningkatkan versi driver minimum yang didukung untuk Greenplum menjadi 42.2.5. Ini juga menambahkan sejumlah perbaikan dan perbaikan bug.

Oktober 18, 2021

[AWS SCT membangun
#1 .0.654](#)

Build 1.0.654 mengimple-
mentasikan konversi skrip
Shell, Teradata FastLoad,
dan Teradata Basic Teradata
Query () ke Amazon Redshift.
BTEQ RSQL Ini juga
menyelesaikan sejumlah
masalah konversi dan
menambahkan sejumlah
perbaikan dan perbaikan bug.

September 16, 2021

[AWS SCT membangun
#1 .0.653](#)

Build 1.0.653 mengimple-
mentasikan konversi dinamis
yang SQL dibuat dalam
fungsi atau prosedur yang
disebut. Ini juga meningkatkan
konversi rutinitas terenkripsi
dan menambahkan sejumlah
perbaikan dan perbaikan bug.

Agustus 10, 2021

[AWS SCT membangun
#1 .0.652](#)

Build 1.0.652 mengimple-
mentasikan mode perintah
skrip di antarmuka baris
perintah dan mengimple-
mentasikan aturan pengoptim-
alan skema. Ini juga
menambahkan sejumlah
konversi dan peningkatan
kinerja dan perbaikan bug.

30 Juni 2021

[AWS SCT membangun
#1 .0.651](#)

Build 1.0.651 menambahk-
an sejumlah perbaikan dan
perbaikan bug. Ini juga
menyediakan akses ke salinan
awal AWS Schema Conversio-
n Tool CLIREferensi.

4 Juni 2021

[AWS SCT membangun
#1 .0.650](#)

Build 1.0.650 mengimple-
mentasikan dukungan Amazon
RDS untuk Postgre SQL 13
sebagai basis data target,
memperbarui agen ekstrakto-
r. Ini juga meningkatkan versi
JDBC driver minimum yang
didukung untuk Microsoft
SQL Server, Azure, dan
Azure Synapse. Selain itu,
ia menambahkan sejumlah
perbaikan konversi dan
perbaikan bug.

30 April 2021

[AWS SCT membangun
#1 .0.649](#)

Build 1.0.649 mengimple-
mentasikan dukungan
MariaDB 10.5 sebagai
database target dan
mengimplementasikan
penyempurnaan fungsi untuk
konversi fungsi bawaan
Oracle. Ini juga menambahk-
an sejumlah konversi dan
peningkatan kinerja dan
perbaikan bug.

29 Maret 2021

[AWS SCT membangun
#1 .0.648](#)

Build 1.0.648 menambahkan
sejumlah perbaikan konversi
dan perbaikan bug.

22 Februari 2021

[AWS SCT membangun
#1 .0.647](#)

Build 1.0.647 menambahkan dukungan fitur Database Mail di AmazonRDS, mengimplementasikan pemuatan dan konversi komentar pada objek penyimpanan. Ini juga menambahkan AWS SCT Data Migration Service Assessor dan AWS SCT Wizard dan mengimplementasikan antarmuka pengguna filter pohon. Selain itu, ia menambahkan bagian yang didesain ulang dalam Laporan Penilaian dan sejumlah perbaikan dan perbaikan bug.

15 Januari 2021

[AWS SCT membangun
#1 .0.646](#)

Build 1.0.646 menambahkan dukungan untuk tipe INTERVAL data, kolom Identity, dan konversi kursor, dan menambahkan sejumlah perbaikan dan perbaikan bug.

28 Desember 2020

[AWS SCT membangun
#1 .0.645](#)

Build 1.0.645 menambahkan dukungan ETL SSIS untuk AWS Glue konversi dan sejumlah perbaikan dan perbaikan bug.

16 November 2020

[AWS SCT membangun
#1 .0.643-1.0.644](#)

Build 1.0.644 menambahkan sejumlah konversi, kinerja, dan peningkatan antarmuka pengguna serta perbaikan bug.

Oktober 14, 2020

AWS SCT membangun #1 .0.642	Build 1.0.642 mengimple mentasikan konversi paket ETL dari Microsoft SQL Server Integration Services ke AWS Glue dan menambahkan sejumlah perbaikan dan perbaikan bug.	28 Agustus 2020
AWS SCT membangun #1 .0.641	Menambahkan SSL dukungan untuk ekstraktor data. Build juga mencakup sejumlah perbaikan dan perbaikan.	17 Juli 2020
AWS SCT membangun #1 .0.633-1.0.640	Diupgrade dari JDK 8 ke Amazon Corretto JDK 11. Menambahkan tabel yang mengidentifikasi peningkatan, perubahan, dan perbaikan lainnya.	22 Juni 2020
AWSWQFketersediaan	AWS SCT tidak lagi menyediakan alat Kerangka Kualifikasi AWS Beban Kerja (AWSWQF) untuk diunduh.	19 Juni 2020
AWS SCT membangun #1 .0.632	SCTUI - Menambahkan tab baru untuk menampilkan kesalahan yang terjadi saat menerapkan skrip. Anda sekarang dapat menyimpan pohon sumber seperti SQL saat mengonversi dari SAPASE. Perbaikan untuk konversi ke Postgre atau SQL Aurora Postgre atau Redshift. SQL	19 November 2019

[AWS SCT membangun #1 .0.631 dan #1 .0.630 \(gabungan\)](#)

Dukungan yang lebih baik ROWIDs di Oracle, dan untuk objek sistem di Microsoft SQL Server dan SAP ASE. Penanganan yang lebih baik untuk penentu skema SQL Server yang hilang. Dukungan yang lebih baik untuk konversi dari Greenplum ke Redshift. Peningkatan dukungan untuk konversi kode yang disimpan saat pindah ke Amazon Redshift, MariaDB, My, dan Postgre. SQL SQL

30 September 2019

[AWS SCT membangun #1 .0.629](#)

Support untuk prosedur tersimpan untuk konversi dari Netezza. Peningkatan dukungan untuk konversi ke Amazon Redshift, DynamoDB, My, dan Postgre. SQL SQL Menambahkan dukungan untuk SAP ASE 12,5 sebagai sumber.

20 Agustus 2019

[AWS SCT membangun #1 .0.628](#)

Support untuk emulasi layanan untuk konversi dari DB2, SQL Server dan Oracle. Penyempurnaan untuk konversi ke Amazon Redshift, termasuk lebih banyak dukungan untuk kursor dan prosedur tersimpan.

22 Juni 2019

<u>AWS SCT membangun #1 .0.627</u>	Support untuk konversi dari SQL Server ke prosedur tersimpan di Amazon Redshift. Perangkat tambahan untuk mengonversi ke Postgre SQL 11 dan My 8.0. SQL	31 Mei 2019
<u>AWS SCT membangun #1 .0.626</u>	Postgre SQL 11 dan My SQL 8.0 sekarang didukung sebagai target. SAPASE15. 5 sekarang didukung sebagai sumber.	26 April 2019
<u>AWS SCT membangun #1 .0.625</u>	Pembaruan mencakup kemampuan untuk mengonversi Teradata ke AWS Glue, dukungan untuk konversi BTEQ ke MariaDB 10.3 dengan dukungan mode kompatibilitas Oracle, dukungan untuk SAP ASE 15.7, dan substitusi layanan untuk meniru fungsionalitas yang hilang.	25 Maret 2019

[AWS SCT membangun
#1 .0.624](#)

Pembaruan mencakup kemampuan untuk mengonversi Oracle ETL ke AWS Glue, dan dukungan untuk konversi dari Microsoft SQL Server, Oracle, dan IBM Db2 ke LUW Amazon untuk MariaDB. RDS Kami juga menambahkan dukungan untuk konversi dari SAP ASE ke RDS untuk My SQL dan Amazon Aurora dengan SQL kompatibilitas Saya. Selain itu, kami menambahkan dukungan untuk ekstensi Oracle selama konversi Oracle ke PostgreSQL.

22 Februari 2019

[AWS SCT membangun
#1 .0.623](#)

Pembaruan mencakup kemampuan untuk mengonversi SAP ASE database, dan kemampuan untuk mengonversi SQL skrip T,DML, dan kode atau komponen yang DDL setara. Kami juga menambahkan emulasi Oracle dan Microsoft SQL Server untuk meningkatkan konversi.

25 Januari 2019

[AWS SCT membangun
#1 .0.622](#)

Pembaruan mencakup Kerangka Kualifikasi Beban Kerja, yang menganalisis beban kerja untuk seluruh migrasi, termasuk modifikasi database dan aplikasi.

20 Desember 2018

AWS SCT membangun #1 .0.621	Pembaruan mencakup dukungan untuk Aurora Postgre SQL 10 sebagai target, dan kemampuan untuk bermigrasi dari Netezza menggunakan opsi tabel eksternal.	21 November 2018
AWS SCT membangun #1 .0.620	Pembaruan mencakup kemampuan untuk menyimpan SQL skrip, dan dukungan untuk kursor global Oracle saat bermigrasi ke My. SQL	22 Oktober 2018
AWS SCT membangun #1 .0.619	Pembaruan termasuk dukungan untuk migrasi dari Apache Cassandra ke DynamoDB, dan dukungan untuk Vertica 9 sebagai sumber.	20 September 2018
AWS SCT membangun #1 .0.618	Pembaruan mencakup laporan penilaian yang diperluas, dukungan untuk mengonversi OracleROWIDs, dan dukungan untuk tabel yang ditentukan pengguna SQL Server.	24 Agustus 2018
AWS SCT membangun #1 .0.617	Pembaruan mencakup laporan penilaian yang diperluas, dukungan untuk mengonversi OracleROWIDs, dan dukungan untuk tabel yang ditentukan pengguna SQL Server.	24 Juli 2018

AWS SCT membangun #1 .0.616	Pembaruan mencakup dukungan untuk RDS saat mengonversi dari Oracle ke Amazon RDS untuk Oracle, mengonversi objek jadwal Oracle, dan dukungan untuk pekerjaan Oracle. partisi, dan Db2 versi 10.1. LUW	26 Juni 2018
AWS SCT membangun #1 .0.615	Pembaruan termasuk dukungan untuk SQL GOTO pernyataan SQL Server ke Postgre, partisi Postgre SQL 10, dan Db2 versi 10.1. LUW	24 Mei 2018
AWS SCT membangun #1 .0.614	Pembaruan termasuk dukungan untuk Oracle ke Oracle DB Links, SQL Server ke Postgre fungsi SQL inline, dan emulasi objek sistem Oracle.	25 April 2018
AWS SCT membangun #1 .0.613	Pembaruan termasuk dukungan untuk Db2LUW, konversi SQL file* Plus, dan SQL Server Windows Authentication.	28 Maret 2018
AWS SCT membangun #1 .0.612	Pembaruan termasuk dukungan untuk pemetaan tipe data kustom, perbandingan skema untuk Oracle 10, dan konversi variabel global Oracle ke PostgreSQL.	22 Februari 2018

[AWS SCT membangun
#1 .0.611](#)

Pembaruan termasuk dukungan untuk pernyataan SQL dinamis Oracle ke Postgre, membuka file log dengan memilih pesan kesalahan, dan kemampuan untuk menyembunyikan skema dalam tampilan pohon.

23 Januari 2018

Pembaruan sebelumnya

Tabel berikut menjelaskan perubahan penting pada AWS Schema Conversion Tool (AWS SCT) panduan pengguna sebelum Januari 2018.

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.608	FIPSDukungan endpoint untuk Amazon S3	Anda sekarang dapat meminta AWS SCT untuk terhubung ke Amazon S3 dan Amazon Redshift dengan FIPS menggunakan titik akhir untuk mematuhi persyaratan keamanan Standar Pemrosesan Informasi Federal. Untuk informasi selengkapnya, lihat Menyimpan AWS kredensi .	17 November 2017
1.0.607	FIPSDukungan endpoint untuk Amazon S3	Anda sekarang dapat meminta AWS SCT untuk terhubung ke Amazon S3 dan Amazon Redshift dengan FIPS menggunakan titik akhir untuk mematuhi persyaratan keamanan Standar Pemrosesan Informasi Federal. Untuk informasi selengkapnya, lihat Menyimpan AWS kredensi .	Oktober 30, 2017
1.0.607	Tugas ekstraksi data dapat	Saat Anda membuat tugas ekstraksi data, Anda sekarang dapat memilih untuk mengabaikan objek besar (LOBs) untuk	Oktober 30, 2017

Versi	Perubahan	Deskripsi	Tanggal diubah
	diabaikan LOBs	mengurangi jumlah data yang Anda ekstrak. Untuk informasi selengkapnya, lihat Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data .	
1.0.605	Akses log tugas agen ekstraksi data	Anda sekarang dapat mengakses log tugas agen ekstraksi data dari tautan yang nyaman di antarmuka AWS Schema Conversion Tool pengguna. Untuk informasi selengkapnya, lihat Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data .	28 Agustus 2017
1.0.604	Penyempurnaan konverter	AWS Schema Conversion Tool Mesin telah ditingkatkan untuk menawarkan konversi yang lebih baik untuk migrasi heterogen.	Juni 24, 2017
1.0.603	Agen ekstraksi data mendukung filter	Anda sekarang dapat memfilter data yang diekstrak agen ekstraksi dari gudang data Anda. Untuk informasi selengkapnya, lihat Membuat aturan migrasi data di AWS SCT .	16 Juni 2017
1.0.603	AWS SCT mendukung versi gudang data tambahan	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengonversi skema Teradata 13 dan Oracle Data Warehouse 10 Anda menjadi skema Amazon Redshift yang setara. Untuk informasi selengkapnya, lihat Mengonversi skema gudang data ke Amazon menggunakan RDS AWS SCT .	16 Juni 2017

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.602	Agen ekstraksi data mendukung gudang data tambahan	Anda sekarang dapat menggunakan agen ekstraksi data untuk mengekstrak data dari gudang data Microsoft SQL Server Anda. Untuk informasi selengkapnya, lihat Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool .	11 Mei 2017
1.0.602	Agen ekstraksi data dapat menyalin data ke Amazon Redshift	Agen ekstraksi data sekarang memiliki tiga mode unggah. Anda sekarang dapat menentukan apakah hanya akan mengekstrak data Anda, untuk mengekstrak data Anda dan hanya mengunggahnya ke Amazon S3, atau untuk mengekstrak, mengunggah, dan menyalin data Anda langsung ke Amazon Redshift. Untuk informasi selengkapnya, lihat Membuat, menjalankan, dan memantau tugas ekstraksi AWS SCT data .	11 Mei 2017
1.0.601	AWS SCT mendukung gudang data tambahan	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengonversi skema Vertica dan Microsoft SQL Server Anda ke skema Amazon Redshift yang setara. Untuk informasi selengkapnya, lihat Mengonversi skema gudang data ke Amazon menggunakan RDS AWS SCT .	April 18, 2017
1.0.601	Agen ekstraksi data mendukung gudang data tambahan	Anda sekarang dapat menggunakan agen ekstraksi data untuk mengekstrak data dari gudang data Greenplum, Netezza, dan Vertica Anda. Untuk informasi selengkapnya, lihat Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool .	April 18, 2017

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.601	Agen ekstraksi data mendukung sistem operasi tambahan	Anda sekarang dapat menginstal agen ekstraksi data di komputer yang menjalankan sistem operasi macOS dan Microsoft Windows. Untuk informasi selengkapnya, lihat Instalasi agen ekstraksi .	April 18, 2017
1.0.601	Agen ekstraksi data mengunggah ke Amazon S3 secara otomatis	Agen ekstraksi data sekarang mengunggah data Anda yang diekstraksi ke Amazon S3 secara otomatis. Untuk informasi selengkapnya, lihat Output tugas ekstraksi data .	April 18, 2017
1.0.600	Agen Ekstraksi Data	Anda sekarang dapat menginstal agen ekstraksi data yang mengekstrak data dari gudang data Anda dan menyiapkannya untuk digunakan dengan Amazon Redshift. Anda dapat menggunakan AWS Schema Conversion Tool untuk mendaftarkan agen dan membuat tugas ekstraksi data untuk mereka. Untuk informasi selengkapnya, lihat Memigrasi data dari gudang data lokal ke Amazon Redshift dengan AWS Schema Conversion Tool .	Februari 16, 2017
1.0.600	Umpan Balik Pelanggan	Anda sekarang dapat memberikan umpan balik tentang AWS Schema Conversion Tool. Anda dapat mengajukan laporan bug, Anda dapat mengirimkan permintaan fitur, atau Anda dapat memberikan informasi umum. Untuk informasi selengkapnya, lihat Memberikan umpan balik .	Februari 16, 2017

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.502	Integrasi dengan AWS DMS	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk membuat AWS DMS titik akhir dan tugas. Anda dapat menjalankan dan memantau tugas dari AWS SCT. Untuk informasi selengkapnya, lihat Integrasi dengan AWS Database Migration Service AWS Schema Conversion Tool .	20 Desember 2016
1.0.502	Amazon Aurora dengan SQL kompatibilitas Postgre sebagai basis data target	AWS Schema Conversion Tool Sekarang mendukung Amazon Aurora dengan SQL kompatibilitas Postgre sebagai basis data target. Untuk informasi selengkapnya, lihat Mengonversi skema database di AWS Schema Conversion Tool .	20 Desember 2016
1.0.502	Support untuk profil	Anda sekarang dapat menyimpan profil yang berbeda di AWS Schema Conversion Tool dan dengan mudah beralih di antara mereka. Untuk informasi selengkapnya, lihat Mengelola Profil di AWS Schema Conversion Tool .	20 Desember 2016
1.0.501	Support untuk Greenplum Database dan Netezza	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengonversi skema gudang data Anda dari Greenplum Database dan Netezza ke Amazon Redshift. Untuk informasi selengkapnya, lihat Mengonversi skema gudang data ke Amazon menggunakan RDS AWS SCT .	17 November 2016

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.501	Optimalisasi pergeseran merah	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengoptimalkan database Amazon Redshift Anda. Untuk informasi selengkapnya, lihat Mengonversi data dari Amazon Redshift menggunakan AWS Schema Conversion Tool .	17 November 2016
1.0.500	Aturan pemetaan	Sebelum Anda mengonversi skema Anda dengan AWS Schema Conversion Tool, Anda sekarang dapat mengatur aturan yang mengubah tipe data kolom, memindahkan objek dari satu skema ke skema lainnya, dan mengubah nama objek. Untuk informasi selengkapnya, lihat Membuat aturan migrasi di AWS SCT .	Oktober 4, 2016
1.0.500	Pindah ke cloud	Sekarang Anda dapat menggunakan AWS Schema Conversion Tool untuk menyalin skema database lokal yang ada ke instans Amazon RDS DB yang menjalankan mesin yang sama. Anda dapat menggunakan fitur ini untuk menganalisis potensi penghematan biaya pindah ke cloud dan mengubah jenis lisensi Anda. Untuk informasi selengkapnya, lihat Menggunakan laporan penilaian di AWS Schema Conversion Tool .	Oktober 4, 2016

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.400	Konversi skema gudang data	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengonversi skema gudang data Anda dari Oracle dan Teradata ke Amazon Redshift. Untuk informasi selengkapnya, lihat Mengonversi skema gudang data ke Amazon menggunakan RDS AWS SCT .	13 Juli 2016
1.0.400	SQLKonversi aplikasi	Anda sekarang dapat menggunakan AWS Schema Conversion Tool untuk mengkonversi SQL dalam C ++, C #, Java, atau kode aplikasi lainnya. Untuk informasi selengkapnya, lihat Mengonversi aplikasi menggunakan SQL AWS SCT .	13 Juli 2016
1.0.400	Fitur baru	AWS Schema Conversion Tool Sekarang berisi paket ekstensi dan wizard untuk membantu Anda menginstal, membuat, dan mengkonfigurasi AWS Lambda fungsi dan pustaka Python untuk menyediakan email, penjadwalan pekerjaan, dan fitur lainnya. Untuk informasi selengkapnya, silakan lihat Menggunakan AWS Lambda fungsi dari paket AWS SCT ekstensi dan Menggunakan pustaka khusus untuk paket AWS SCT ekstensi .	13 Juli 2016
1.0.301	SSLSupport	Anda sekarang dapat menggunakan Secure Sockets Layer (SSL) untuk terhubung ke database sumber Anda saat Anda menggunakan file AWS Schema Conversion Tool.	Mei 19, 2016

Versi	Perubahan	Deskripsi	Tanggal diubah
1.0.203	Fitur baru	Menambahkan dukungan untuk My SQL dan Postgre SQL sebagai database sumber untuk konversi.	11 April 2016
1.0.202	Rilis pemeliharaan	Menambahkan dukungan untuk mengedit konversi SQL yang dihasilkan untuk mesin database target. Menambahkan kemampuan seleksi yang ditingkatkan dalam database sumber dan menargetkan tampilan pohon instans DB. Menambahkan dukungan untuk menghubungkan ke database sumber Oracle menggunakan nama Transparent Network Substrate (TNS).	2 Maret 2016
1.0.200	Rilis pemeliharaan	Menambahkan dukungan untuk Postgre SQL sebagai mesin database target. Menambahkan kemampuan untuk menghasilkan skema yang dikonversi sebagai skrip dan menyimpan skrip ke file sebelum menerapkan skema ke instans DB target.	14 Januari 2016
1.0.103	Rilis pemeliharaan	Menambahkan kemampuan proyek offline, kemampuan untuk memeriksa versi baru, dan manajemen memori dan kinerja.	Desember 2, 2015
1.0.101	Rilis pemeliharaan	Menambahkan wizard Buat Proyek Migrasi Database Baru. Menambahkan kemampuan untuk menyimpan laporan penilaian migrasi database sebagai PDF file.	19 Oktober 2015
1.0.100	Rilis pratinjau	Menyediakan panduan pengguna untuk rilis AWS Schema Conversion Tool pratinjau.	7 Oktober 2015

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.