Guía del usuario

AWS Schema Conversion Tool

Version 1.0.672

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

AWS Schema Conversion Tool: Guía del usuario

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

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

Table of Contents

¿Qué es AWS SCT? 1
Introducción a la conversión de esquemas 5
Ofrecer comentarios 7
Instalar, verificar y actualizar 8
Instalando AWS SCT 8
Instalar versiones anteriores 10
Verificar la descarga del AWS SCT archivo 10
Verificar la suma de comprobación del archivo AWS SCT 11
Verificación de los archivos AWS SCT RPM en Fedora 11
Verificación de los archivos AWS SCT DEB en Ubuntu 12
Verificación del archivo AWS SCT MSI en Microsoft Windows 13
Descarga de los controladores de base de datos necesarios 13
Instala controladores JDBC en Linux 17
Almacenamiento de rutas a los controladores en la configuración global 18
Actualizando AWS SCT 20
AWS SCT CLI 20
Utilización de la interfaz de usuario de AWS SCT 21
La ventana de proyecto 21
Inicio de AWS SCT 23
Creación de un proyecto 23
Uso de un asistente de proyecto nuevo 24
Cómo guardar y abrir un proyecto 27
Cómo agregar un servidor 28
Uso de un modo sin conexión 29
Utilización de los filtros de árbol 30
..... 31
Importación de una lista de archivos para el filtro de árbol 33
Cómo ocultar esquemas 34
Administración del informe de evaluación de la migración de la base de datos 36
Conversión de un esquema 40
Aplicación del código convertido 43
Almacenamiento de perfiles de AWS 44
Almacenamiento de credenciales de AWS 44
Configurar el perfil predeterminado para un proyecto 47

Permisos para usar el perfil de servicio de AWS ... 47
Utilizar AWS Secrets Manager ... 48
Almacenar contraseñas de la base de datos ... 49
Utilizar la vista Union All en proyectos con tablas con particiones ... 49
Métodos abreviados de teclado ... 50
Introducción ... 52
Orígenes para AWS SCT ... 54
Cifrado de las conexiones de Amazon RDS ... 55
Usar Apache Cassandra como origen ... 58
Conexión a Apache Cassandra como origen ... 58
Usar Apache Hadoop como origen ... 60
Requisitos previos para utilizar Apache Hadoop como origen ... 61
Permisos para Hive como origen ... 62
Permisos para HDFS como origen ... 62
Permisos para HDFS como destino ... 63
Conexión a Apache Hadoop como origen ... 63
Conexión a Hive y HDFS ... 65
Conexión a Amazon EMR como destino ... 68
Usar Apache Oozie como origen ... 71
Requisitos previos ... 71
Conexión a Apache Oozie como origen ... 72
Permisos para AWS Lambda ... 73
Conexión a AWS Step Functions como origen ... 76
Usar Azure SQL Database como origen ... 77
Privilegios para Azure SQL Database ... 77
Conectar a Azure SQL Database como origen ... 78
Usar IBM Db2 para z/OS como origen ... 79
Requisitos previos para Db2 for z/OS ... 80
Privilegios para Db2 para z/OS ... 80
Conectar a Db2 para z/OS como origen ... 82
Privilegios para MySQL como destino ... 84
Privilegios para PostgreSQL como destino ... 86
Configuración de conversión de Db2 para z/OS a PostgreSQL ... 86
Usar IBM Db2 LUW como origen ... 88
Privilegios para Db2 para LUW ... 89
Conectar a Db2 LUW como origen ... 91

De Db2 LUW a PostgreSQL 93
De Db2 LUW a MySQL 96
Usar MySQL como origen 97
Privilegios para MySQL 98
Conectar a MySQL como origen 98
Privilegios para PostgreSQL como destino 101
Usar una base de datos de Oracle como origen 101
Privilegios para Oracle 102
Conexión a Oracle como origen 102
Oracle a PostgreSQL 106
De Oracle a MySQL 112
De Oracle a Amazon RDS para Oracle 122
Uso de PostgreSQL como origen 129
Privilegios para PostgreSQL 129
Conectar a PostgreSQL como origen 130
Privilegios para MySQL como destino 133
Usar SAP ASE (Sybase ASE) como origen 134
Privilegios para SAP ASE 135
Conectar a SAP ASE (Sybase) como origen 135
Privilegios para MySQL como destino 138
Configuración de conversión de SAP ASE a MySQL 139
Privilegios para PostgreSQL como destino 140
Configuración de conversión de SAP ASE a PostgreSQL 141
Uso de SQL Server como origen 142
Privilegios de Microsoft SQL Server 143
Usar la autenticación de Windows con Microsoft SQL Server 144
Conectar a SQL Server como origen 146
De SQL Server a MySQL 149
De SQL Server a PostgreSQL 154
De SQL Server a Amazon RDS SQL Server 191
Orígenes de almacenamiento de datos para AWS SCT 192
Uso de Amazon Redshift como origen 193
Utilizar Azure Synapse Analytics como origen para 199
Usar BigQuery como origen 205
Utilizar la base de datos de Greenplum como origen 211
Utilizar Netezza como origen 217

Utilizar Oracle Data Warehouse como origen 227
Utilizar Snowflake como origen 235
Utilizar Microsoft SQL Server Data Warehouse como origen 244
Utilizar Teradata como origen 251
Uso de Vertica como origen 267
Creación de reglas de asignación 274
Regla nueva 275
Administración de reglas 275
Destinos virtuales 277
Limitaciones 278
Creación de informes de conversión 279
Informes de evaluación de migración 279
Creación de un informe de evaluación de la migración de la base de datos 280
Ver el informe de evaluación 281
Guardar el informe de evaluación 285
Configurar el informe de evaluación 287
Crear un informe de evaluación multiservidor 291
Convertir esquemas de bases de datos 301
Crear reglas de migración 303
Creación de reglas de migración 304
Cómo exportar reglas de migración 306
Convertir su esquema 306
Convertir esquemas 307
Editar esquemas convertidos 310
Eliminar un esquema convertido 311
Administrar conversiones manuales 311
Modificar su esquema de origen 311
Modificar su esquema de destino 312
Actualizar un esquema convertido 312
Guardar y aplicar un esquema 313
Cómo guardar el esquema convertido 313
Aplicación del esquema convertido 314
Esquema del paquete de extensión 315
Comparación de esquemas 315
Objetos transformados relacionados 317
Convertir esquemas de almacenamiento de datos a Amazon Redshift 318

Permisos para Amazon Redshift 319
Seleccionar estrategias y reglas de optimización 321
Recopilar o cargar estadísticas 322
Crear reglas de migración 324
Crear reglas de migración 325
Exportar reglas de migración 327
Convertir su esquema 327
Convertir esquemas 328
Editar el esquema convertido 330
Eliminar un esquema convertido 331
Administrar y personalizar claves 331
Temas relacionados 332
Crear y usar el informe de evaluación 332
Crear un informe de evaluación de la migración de la base de datos 332
Resumen 333
Elementos de acción 335
Guardar el informe de evaluación 335
Administrar conversiones manuales 336
Modificar su esquema de origen 337
Modificar su esquema de destino 337
Actualizar un esquema convertido 338
Guardar y aplicar un esquema convertido 338
Guardar el esquema convertido en un archivo 339
Aplicar el esquema convertido 339
El esquema del paquete de extensión 340
Bibliotecas Python 341
Optimizar Amazon Redshift 341
Optimizar la base de datos de Amazon Redshift 341
Conversión de procesos ETL 344
Convertir procesos de ETL a AWS Glue 345
Requisitos previos 346
AWS Glue Data Catalog 347
Limitaciones 347
Paso 1: Crear un nuevo proyecto de 349
Paso 2: Crear un trabajo de AWS Glue 350
Convertir procesos de ETL mediante la API de Python para AWS Glue 351

Paso 1: Crear una base de datos ... 352
Paso 2: Crear una conexión ... 352
Paso 3: Crear un rastreador de AWS Glue ... 354
Convertir los scripts de ETL de Informatica ... 356
Conversión SSIS a AWS Glue ... 361
Componentes de SSIS admitidos ... 365
Convertir SSIS a AWS Glue Studio ... 367
Requisitos previos ... 368
Añadir paquetes de SSIS a su proyecto de AWS SCT ... 370
Conversión de paquetes de SSIS ... 371
Crear trabajos de AWS Glue Studio ... 371
Creación de un informe de evaluación de la conversión de SSIS ... 373
Componentes de SSIS admitidos ... 374
Convertir Teradata BTEQ a RSQL de Amazon Redshift ... 375
Agregar scripts de BTEQ a su proyecto de AWS SCT ... 377
Configurar las variables de sustitución en los scripts de BTEQ ... 378
Conversión de scripts de BTEQ ... 379
Administrar scripts de BTEQ ... 379
Crear un informe de evaluación de la conversión de scripts de BTEQ ... 380
Editar y guardar scripts de BTEQ convertidos ... 381
Convertir scripts del intérprete de comandos a RSQL de Amazon Redshift ... 381
Agregar scripts de intérprete de comandos a su proyecto de AWS SCT ... 382
Configurar variables de sustitución en scripts de intérprete de comandos ... 383
Convertir scripts del intérprete de comandos ... 384
Administrar scripts del intérprete de comandos ... 385
Crear un informe de evaluación de la conversión de scripts del intérprete de comandos ... 386
Editar y guardar scripts del intérprete de comandos convertidos ... 387
Convertir de FastExport de Teradata a RSQL de Amazon Redshift ... 387
Agregar scripts de trabajo de FastExport a su proyecto de AWS SCT ... 388
Configurar las variables de sustitución en los scripts de trabajo de FastExport de Teradata ... 389
Convertir scripts de trabajo de FastExport ... 390
Administración de scripts de trabajo de FastExport ... 391
Crear un informe de evaluación de la conversión de scripts de trabajo de FastExport de Teradata ... 391
Editar y guardar los scripts de trabajo de FastExport convertidos ... 392

Convertir FastExport de Teradata a RSQL de Amazon Redshift ... 393
Cómo agregar scripts de trabajo de FastLoad al proyecto de AWS SCT ... 394
Configurar las variables de sustitución en los scripts de trabajo de FastLoad de Teradata ... 395
Conversión de scripts de trabajo de FastLoad ... 396
Administración de scripts de trabajo de FastLoad ... 397
Crear un informe de evaluación de la conversión de scripts de trabajo de FastLoad de Teradata ... 398
Editar y guardar los scripts de trabajo de FastLoad convertidos ... 399
Convertir scripts de trabajo de MultiLoad de Teradata a RSQL de Amazon Redshift ... 399
Cómo agregar scripts de trabajo de MultiLoad a su proyecto de AWS SCT ... 400
Configurar las variables de sustitución en los scripts de trabajo de MultiLoad ... 401
Conversiones de scripts de trabajo de MultiLoad ... 402
Administración de scripts de trabajo de MultiLoad ... 403
Crear un informe de evaluación de la conversión de scripts de trabajo de MultiLoad de Teradata ... 404
Editar y guardar los scripts de trabajo de MultiLoad convertidos ... 405
Migración de marcos de macrodatos ... 406
Migración de Apache Hadoop a Amazon EMR con ... 406
Información general ... 407
Paso 1: Conectarse a los clústeres de Hadoop ... 408
Paso 2: Configurar las reglas de asignación ... 408
Paso 3: Crear un informe de evaluación ... 410
Paso 4: Migrar su clúster de Apache Hadoop a Amazon EMR con ... 411
Ejecución del script de la CLI ... 412
Administración del proyecto de migración ... 412
Conversión de Apache Oozie a AWS Step Functions ... 414
Información general ... 415
Paso 1: Conectar con los servicios de origen y destino ... 416
Paso 2: Configurar las reglas de asignación ... 417
Paso 3: Configurar los parámetros ... 418
Paso 4: Crear un informe de evaluación ... 419
Paso 5: Convertir los flujos de trabajo de Apache Oozie a AWS Step Functions ... 421
Ejecución del script de la CLI ... 424
Nodos compatibles ... 424
Uso de AWS SCT con AWS DMS ... 426
Uso de un agente de replicación de AWS SCT con AWS DMS ... 426

Uso de un agente de extracción de datos de AWS SCT con AWS DMS 426
Cómo aumentar los niveles de registro cuando se usa AWS SCT con AWS DMS 426
Migración de un almacén de datos a Amazon Redshift 428
Requisitos previos 431
Configuración de Amazon S3 431
Asunción de roles de IAM 432
Configuración de seguridad 434
Opciones de configuración 434
Instalación de agentes 435
Configuración de agentes 437
Instalación y configuración de agentes de extracción con agentes de copia dedicados 439
Inicio de agentes 440
Registro de agentes 441
Ocultar y recuperar la información de un AWS SCT agente 441
Creación de reglas de migración de datos 443
Cambio de los ajustes de extracción y copia para la migración de datos 444
Ordenación de datos 448
Crear, ejecutar y supervisar una AWS SCT tarea 449
Exportación e importación de una tarea de extracción de datos 453
Extracción de datos mediante un dispositivo AWS Snowball Edge 454
tep-by-step Procedimientos S para migrar datos mediante Edge AWS SCT AWS Snowball 455
Resultado de la tarea de extracción de datos 458
Uso de particiones virtuales 460
Límites del particionamiento virtual 460
Tipo de partición RANGE 460
Tipo de partición LIST 462
Tipo de partición DATE AUTO SPLIT 463
Uso de particiones nativas 464
Trabajo con LOB 465
Prácticas recomendadas y solución de problemas 466
Convertir el SQL de las aplicaciones 468
Información general de la conversión del SQL de las aplicaciones 468
Conversión del código SQL de sus aplicaciones 469
Crear proyectos de conversión genérica de aplicaciones 469
Administrar proyectos de conversión de aplicaciones 473

Analizar y convertir su código SQL ... 474
Crear y usar el informe de evaluación ... 475
Editar y guardar el código SQL convertido ... 477
Convertir el código SQL de aplicaciones de C# ... 477
Crear proyectos de conversión de aplicaciones de C# ... 477
Convertir el código SQL de la aplicación de C# ... 479
Cómo guardar el código de la aplicación convertida ... 480
Administrar proyectos de conversión de aplicaciones de C# ... 481
Crear un informe de evaluación de la conversión de la aplicación de C# ... 482
Convertir el código SQL de aplicaciones de C++ ... 483
Crear proyectos de conversión de aplicaciones de C++ ... 483
Convertir el código SQL de la aplicación de C++ ... 485
Cómo guardar el código de la aplicación convertida ... 486
Administrar proyectos de conversión de aplicaciones de C++ ... 487
Crear un informe de evaluación de la conversión de la aplicación de C++ ... 489
Convertir el código SQL en aplicaciones de Java ... 490
Crear proyectos de conversión de aplicaciones de Java ... 490
Convertir el código SQL de la aplicación de Java ... 492
Cómo guardar el código de la aplicación convertida ... 494
Administrar proyectos de conversión de aplicaciones de Java ... 494
Crear un informe de evaluación de la conversión de la aplicación de Java ... 495
Conversión del código SQL en aplicaciones de Pro*C ... 497
Creación de proyectos de conversión de aplicaciones de Pro*C ... 497
Convertir el código SQL de la aplicación de Pro*C ... 499
Editar y guardar el código de la aplicación convertida ... 500
Administrar proyectos de conversión de aplicaciones de Pro*C ... 501
Crear un informe de evaluación de la conversión de la aplicación de Pro*C ... 502
Uso de paquetes de extensión ... 504
Permisos para usar el paquete de extensión ... 505
Uso del esquema del paquete de extensión ... 507
Bibliotecas personalizadas para los paquetes de extensión ... 507
Aplicación del paquete de extensión ... 508
Uso de las funciones de Lambda del paquete de extensión de AWS SCT ... 510
Uso de AWS Lambda funciones para emular la funcionalidad de la base de datos ... 510
Aplicación del paquete de extensión para admitir funciones de Lambda ... 511
Configuración de las funciones del paquete de extensión ... 512

Prácticas recomendadas 514
Configurar memoria adicional 514
Carpeta del proyecto predeterminada 514
Incrementación de la velocidad de migración de los datos 515
Cómo aumentar la información de registro 515
Solución de problemas 519
No se puede cargar objetos de una base de datos de origen de Oracle 519
Mensaje de advertencia 519
Referencia de la CLI 521
Requisitos previos 521
Modo interactivo 521
Ejemplos 523
Obtención de escenarios de la CLI 523
Ejemplos 528
Edición de escenarios de la CLI 528
Modo script 529
Ejemplos 530
Material de referencia 531
Notas de la versión 532
Notas de versión: 675 532
Notas de la versión – 674 534
Notas de la versión – 673 542
Notas de la versión – 672 547
Notas de la versión – 671 555
Notas de la versión – 670 564
Notas de la versión – 669 569
Notas de la versión – 668 575
Notas de la versión – 667 581
Notas de la versión – 666 585
Notas de la versión – 665 590
Notas de la versión – 664 593
Notas de la versión – 663 597
Notas de la versión – 662 600
Notas de la versión – 661 605
Notas de la versión – 660 610
Notas de la versión – 659 614

Notas de la versión – 658 618
Notas de la versión – 657 623
Notas de la versión – 656 628
Notas de la versión – 655 631
Notas de la versión – 654 634
Notas de la versión – 653 637
Notas de la versión – 652 639
Notas de la versión – 651 642
Notas de la versión – 650 644
Notas de la versión – 649 646
Notas de la versión – 648 649
Notas de la versión – 647 650
Notas de la versión – 646 652
Notas de la versión – 645 654
Notas de la versión – 644 655
Notas de la versión – 642 658
Notas de la versión – 641 659
Notas de la versión – 640 660
Cambios en la versión 1.0.640 de Oracle 660
Cambios en la versión 1.0.640 de Microsoft SQL Server 667
Cambios en la versión 1.0.640 de MySQL 671
Cambios en la versión 1.0.640 de PostgreSQL 672
Cambios en la versión 1.0.640 de Db2 LUW 675
Cambios en la versión 1.0.640 de Teradata 676
Cambios en la versión 1.0.640 para otros motores 678
Historial de documentos 681
Actualizaciones anteriores 699
..... dccviii

¿Qué es la AWS Schema Conversion Tool?

Puede utilizar la AWS Schema Conversion Tool (AWS SCT) para convertir su esquema de base de datos existente de un motor de base de datos a otro. Puede convertir esquemas relacionales OLTP o esquemas de data warehouse. El esquema convertido es adecuado para una base de datos MySQL de Amazon Relational Database Service (Amazon RDS), MariaDB, Oracle, SQL Server o PostgreSQL, un clúster de base de datos de Amazon Aurora o un clúster de Amazon Redshift. El esquema convertido también puede utilizarse con una base de datos en una instancia Amazon EC2 o almacenarse como datos en un bucket de Amazon S3.

AWS SCT admite varios estándares del sector, incluido el Estándar federal de procesamiento de la información (FIPS, Federal Information Processing Standards) para conexiones a un bucket de Amazon S3 o cualquier otro recurso de AWS. AWS SCT también cumple con el Programa federal de administración de riesgos y autorizaciones (FedRAMP, Federal Risk and Authorization Management Program). Para obtener más detalles sobre AWS y los esfuerzos de conformidad, consulte Servicios de AWS en el alcance por programa de conformidad.

AWS SCT es compatible con las siguientes conversiones de OLTP.

Base de datos de origen	Bases de datos de destino
IBM Db2 para z/OS (versión 12)	Amazon Aurora MySQL-Compatible Edition (Aurora MySQL), Amazon Aurora PostgreSQL-Compatible Edition (Aurora PostgreSQL), MySQL, PostgreSQL Para obtener más información, consulte Usar IBM Db2 para z/OS como origen.
IBM Db2 LUW (versiones 9.1, 9.5, 9.7, 10.5, 11.1 y 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL Para obtener más información, consulte Usar IBM Db2 LUW como origen.
Microsoft Azure SQL Database	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL

Base de datos de origen	Bases de datos de destino
	Para obtener más información, consulte Usar Azure SQL Database como origen.
Microsoft SQL Server (versión 2008 R2, 2012, 2014, 2016, 2017, 2019 y 2022)	Aurora MySQL, Aurora PostgreSQL, Babelfish para Aurora PostgreSQL (solo para informes de evaluaciones), MariaDB, Microsoft SQL Server, MySQL, PostgreSQL Para obtener más información, consulte Uso de SQL Server como origen.
MySQL (versión 5.5 y posteriores)	Aurora PostgreSQL, MySQL, PostgreSQL Para obtener más información, consulte Usar MySQL como origen. Puede migrar esquemas y datos de MySQL a un clúster de base de datos de Aurora MySQL sin usar AWS SCT. Para obtener más información, consulte Migración de datos a un clúster de base de datos de Amazon Aurora.
Oracle (versión 10.1 y posteriores)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL Para obtener más información, consulte Usar una base de datos de Oracle como origen.
PostgreSQL (versión 9.1 y posteriores)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL Para obtener más información, consulte Uso de PostgreSQL como origen.

Base de datos de origen	Bases de datos de destino
SAP ASE (versiones 12.5.4, 15.0.2, 15.5, 15.7 y 16.0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL Para obtener más información, consulte Usar SAP ASE (Sybase ASE) como origen.

AWS SCT es compatible con las siguientes conversiones de data warehouse.

Almacenamiento de datos de origen	Almacenamiento de datos de destino
Amazon Redshift	Amazon Redshift Para obtener más información, consulte Uso de Amazon Redshift como origen.
Azure Synapse Analytics	Amazon Redshift Para obtener más información, consulte Utilizar Azure Synapse Analytics como origen para .
BigQuery	Amazon Redshift Para obtener más información, consulte Usar BigQuery como origen.
Greenplum Database (versiones 4.3 y 6.21)	Amazon Redshift Para obtener más información, consulte Utilizar la base de datos de Greenplum como origen.
Microsoft SQL Server (versión 2008 y posteriores)	Amazon Redshift Para obtener más información, consulte Utilizar Microsoft SQL Server Data Warehouse como origen.
Netezza (versión 7.0.3 y posteriores)	Amazon Redshift

Almacenamiento de datos de origen	Almacenamiento de datos de destino
	Para obtener más información, consulte Utilizar Netezza como origen.
Oracle (versión 10.1 y posteriores)	Amazon Redshift Para obtener más información, consulte Utilizar Oracle Data Warehouse como origen.
Snowflake (versión 3)	Amazon Redshift Para obtener más información, consulte Utilizar Snowflake como origen.
Teradata (versión 13 y posteriores)	Amazon Redshift Para obtener más información, consulte Utilizar Teradata como origen.
Vertica (versión 7.2.2 y posteriores)	Amazon Redshift Para obtener más información, consulte Uso de Vertica como origen.

AWS SCT es compatible con las siguientes conversiones de bases de datos de NoSQL.

Base de datos de origen	Bases de datos de destino
Apache Cassandra (versiones 2.1.x, 2.2.16 y 3.11.x)	Amazon DynamoDB Para obtener más información, consulte Usar Apache Cassandra como origen.

AWS SCT admite conversiones de los siguientes procesos de extracción, transformación y carga (ETL). Para obtener más información, consulte Conversión de procesos ETL.

Origen	Destino
Scripts de ETL de Informatica	Informatica
Paquetes de ETL de Microsoft SQL Server Integration Services (SSIS)	AWS Glue o AWS Glue Studio
Scripts de intérprete de comandos con comandos incrustados de Teradata Basic Teradata Query (BTEQ)	Amazon Redshift RSQL
Scripts de ETL de Teradata BTEQ	RSQL de AWS Glue o Amazon Redshift
Scripts de trabajo de Teradata FastExport	Amazon Redshift RSQL
Scripts de trabajo de Teradata FastLoad	Amazon Redshift RSQL
Scripts de trabajo de Teradata MultiLoad	Amazon Redshift RSQL

AWS SCT admite las siguientes migraciones de marcos de macrodatos. Para obtener más información, consulte Migración de marcos de macrodatos.

Origen	Destino
Apache Hive (versión 0.13.0 y posteriores)	Hive en Amazon EMR
Apache HDFS	Amazon S3 o HDFS en Amazon EMR
Apache Oozie	AWS Step Functions

Introducción a la conversión de esquemas

AWS SCT proporciona una interfaz de usuario basada en el proyecto para convertir automáticamente el esquema de la base de datos de origen en un formato compatible con su instancia de Amazon RDS de destino. Si el esquema de la base de datos de origen no se puede convertir automáticamente, AWS SCT facilita orientación sobre cómo puede crear un esquema equivalente en su base de datos de destino.

Para obtener información acerca de cómo instalar la AWS SCT, consulte Instalación, verificación y actualización AWS SCT.

Para obtener una introducción a la interfaz de usuario de AWS SCT, consulte Utilización de la interfaz de usuario de AWS SCT.

Para obtener información acerca del proceso de conversión, consulte Conversión de esquemas de base de datos mediante AWS SCT.

Además de convertir su esquema de base de datos existente de un motor de base de datos a otro, AWS SCT tiene algunas características adicionales que le ayudarán a trasladar sus datos y aplicaciones a la nube de AWS:

- Puede utilizar agentes de extracción de datos para extraer datos de su almacenamiento de datos para prepararlos y migrarlos a Amazon Redshift. Para administrar los agentes de extracción de datos, puede utilizar la AWS SCT. Para obtener más información, consulte Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift.
- Puede utilizar la AWS SCT para crear puntos de conexión y tareas de AWS DMS. Puede ejecutar y monitorizar estas tareas desde la AWS SCT. Para obtener más información, consulte Uso de AWS SCT con AWS DMS.
- En algunos casos, las características de la base de datos de origen no se pueden convertir a características de Amazon Redshift equivalentes. El asistente de paquete de extensión de la AWS SCT puede ayudarle a instalar funciones de AWS Lambda y bibliotecas Python para emular las características que no se puede convertir. Para obtener más información, consulte Uso de paquetes AWS SCT de extensión.
- Puede utilizar AWS SCT para optimizar su base de datos de Amazon Redshift existente. AWS SCT recomienda claves de clasificación y distribución para optimizar su base de datos. Para obtener más información, consulte Optimizar Amazon Redshift mediante AWS SCT.
- También puede utilizar AWS SCT para copiar un esquema de base de datos en las instalaciones existente a una instancia de base de datos de Amazon RDS que ejecute el mismo motor. Puede utilizar esta característica para analizar posibles ahorros de costos en la migración a la nube y en el cambio del tipo de licencia.
- Puede utilizar la AWS SCT para convertir SQL en su código C++, C #, Java u otro tipo de código de aplicación. Puede ver, analizar, editar y guardar el código SQL convertido. Para obtener más información, consulte Conversión de SQL de las aplicaciones mediante AWS SCT.

- Puede utilizar AWS SCT para migrar procesos de extracción, transformación y carga (ETL). Para obtener más información, consulte Conversión de procesos de extracción, transformación y carga (ETL) con AWS Schema Conversion Tool.

Enviar comentarios

Puede ofrecer sus comentarios sobre AWS SCT. Puede archivar un informe de errores, enviar una solicitud de característica o proporcionar información general.

Para ofrecer sus comentarios sobre AWS SCT

1. Inicie AWS Schema Conversion Tool.
2. Abra el menú Help y seleccione Leave Feedback. Aparecerá el cuadro de diálogo Leave Feedback.
3. En Area, seleccione Information, Bug report o Feature request.
4. En Source database, seleccione su base de datos de origen. Seleccione Any si sus comentarios no se refieren a una base de datos en particular.
5. En Target database, seleccione su base de datos de destino. Seleccione Any si sus comentarios no se refieren a una base de datos en particular.
6. En Title, escriba un título para sus comentarios.
7. En Message, escriba sus comentarios.
8. Seleccione Send para enviar sus comentarios.

Instalación, verificación y actualización AWS SCT

The AWS Schema Conversion Tool (AWS SCT) es una aplicación independiente que proporciona una interfaz de usuario basada en proyectos. AWS SCT está disponible para Microsoft Windows, Fedora Linux y Ubuntu Linux. AWS SCT solo es compatible con sistemas operativos de 64 bits.

Para asegurarnos de que obtiene la versión correcta del archivo de AWS SCT distribución, proporcionamos los pasos de verificación después de descargar el archivo comprimido. Puede verificar el archivo por medio de los pasos proporcionados.

Temas

- ¿Instalando AWS SCT
- Verificar la descarga del AWS SCT archivo
- Descarga de los controladores de base de datos necesarios
- Actualizando AWS SCT
- AWS SCT CLI

¿Instalando AWS SCT

Puede instalarlo AWS SCT en los siguientes sistemas operativos:

- Microsoft Windows 10
- Fedora Linux 36 y posterior
- Ubuntu Linux 18 y posterior

Para instalar AWS SCT

1. Descargue el archivo comprimido que contiene el AWS SCT instalador mediante el enlace de su sistema operativo. Todos los archivos comprimidos tienen la extensión .zip. Cuando extraiga el archivo de AWS SCT instalación, tendrá el formato adecuado para su sistema operativo.

    - Microsoft Windows
    - Ubuntu Linux (.deb)
    - Fedora Linux (.rpm)

2. Extraiga el archivo de AWS SCT instalación de su sistema operativo, que se muestra a continuación.

Sistema operativo	Nombre de archivo
Fedora Linux	`aws-schema-conversion-tool-1.0.` *`build-number`* `.x86_64.rpm`
Microsoft Windows	`AWS Schema Conversion Tool-1.0.` *`build-number`* `.msi`
Ubuntu Linux	`aws-schema-conversion-tool-1.0.` *`build-number`* `.deb`

3. Ejecute el archivo de AWS SCT instalación extraído en el paso anterior. Utilice las instrucciones para su sistema operativo, que se muestran a continuación.

Sistema operativo	Instrucciones de instalación
Fedora Linux	Ejecute el siguiente comando en la carpeta en la que haya descargado el archivo: `sudo yum install aws-schema-conversion-tool-1.0.` *`build-number`* `.x86_64.rpm`
Microsoft Windows	Haga doble clic en el archivo para ejecutar el instalador.
Ubuntu Linux	Ejecute el siguiente comando en la carpeta en la que haya descargado el archivo: `sudo dpkg -i aws-schema-conversion-tool-1.0.` *`build-number`* `.deb`

4. Descargue los controladores de Java Database Connectivity (JDBC) para sus motores de base de datos de origen y destino. Para obtener instrucciones y enlaces de descarga, consulte Descarga de los controladores de base de datos necesarios.

Ahora, ha completado la configuración de la AWS SCT aplicación. Haga doble clic en el icono de la aplicación para ejecutar AWS SCT.

Instalación de versiones anteriores de AWS SCT

Puede descargar e instalar versiones anteriores a AWS SCT partir de la versión 1.0.625. Para descargar una versión anterior, proporcione la información sobre la versión y el sistema operativo con el siguiente formato.

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

Por ejemplo, para descargar la AWS SCT versión 668, haga lo siguiente:

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

AWS es compatible con las tres versiones más recientes de la aplicación. AWS SCT Le recomendamos que utilice la última versión de AWS SCT.

Verificar la descarga del AWS SCT archivo

Hay varias formas de comprobar el archivo de distribución de AWS SCT. La más sencilla es comparar la suma de comprobación del archivo con la suma de comprobación publicada de AWS. Como nivel de seguridad adicional, puede utilizar los procedimientos que se muestran a continuación, para verificar el archivo de distribución, en función del sistema operativo en el que instaló el archivo.

Esta sección incluye los siguientes temas.

Temas

- Verificar la suma de comprobación del archivo AWS SCT
- Verificación de los archivos AWS SCT RPM en Fedora
- Verificación de los archivos AWS SCT DEB en Ubuntu
- Verificación del archivo AWS SCT MSI en Microsoft Windows

Verificar la suma de comprobación del archivo AWS SCT

Para detectar cualquier error que se haya podido introducir al descargar o almacenar el archivo AWS SCT comprimido, puede comparar la suma de comprobación del archivo con un valor proporcionado por. AWS AWS utiliza el algoritmo SHA256 para la suma de comprobación.

Para verificar el archivo de AWS SCT distribución mediante una suma de comprobación

1. Descargue el archivo AWS SCT de distribución mediante los enlaces de la sección de instalación. Para obtener más información, consulte ¿Instalando AWS SCT.
2. Descargue el archivo de suma de comprobación, llamado sha256Check.txt. Este archivo incluye las sumas de comprobación de la última AWS SCT versión. Por ejemplo, el archivo puede tener un aspecto como este:

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

3. Ejecute el comando de validación SHA256 para su sistema operativo en el directorio que contiene el archivo de distribución. Por ejemplo, ejecute el siguiente comando en Linux.

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

4. Compare los resultados del comando con el valor se muestra en el archivo sha256Check.txt. Si las sumas de comprobación coinciden, entonces es seguro ejecutar el archivo de distribución. Si las sumas de comprobación no coinciden, no ejecute el archivo de distribución y póngase en contacto con AWS Support.

Verificación de los archivos AWS SCT RPM en Fedora

AWS proporciona otro nivel de validación además de la suma de comprobación del archivo de distribución. Todos los archivos RPM del archivo de distribución están firmados por una clave AWS privada. La clave pública de GPG se puede consultar en amazon.com.public.gpg-key.

Para verificar los archivos AWS SCT RPM en Fedora

1. Descargue el archivo AWS SCT de distribución mediante los enlaces de la sección de instalación.
2. Compruebe la suma de comprobación del archivo de AWS SCT distribución.
3. Extraiga el contenido del archivo de distribución. Localice el archivo RPM que desea verificar.
4. Descargue la clave pública de GPG desde amazon.com.public.gpg-key
5. Importe la clave pública a su base de datos RPM (asegúrese de obtener los permisos pertinentes) con el siguiente comando:

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

6. Para comprobar que la importación se ha realizado correctamente, ejecute el siguiente comando:

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

7. Para comprobar la firma de RPM, ejecute el siguiente comando:

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

Verificación de los archivos AWS SCT DEB en Ubuntu

AWS proporciona otro nivel de validación además de la suma de comprobación del archivo de distribución. Todos los archivos DEB en el archivo de distribución están firmados por una firma separada de GPG.

Para comprobar los archivos AWS SCT DEB en Ubuntu

1. Descargue el archivo AWS SCT de distribución mediante los enlaces de la sección de instalación.
2. Verificar la suma de comprobación del archivo de AWS SCT distribución.
3. Extraiga el contenido del archivo de distribución. Localice el archivo DEB que desea verificar.
4. Descargue la firma separada de -1.0.latest.deb.ascaws-schema-conversion-tool.
5. Descargue la clave pública de GPG desde amazon.com.public.gpg-key.

6. Para importar la clave GPG pública, ejecute el siguiente comando:

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

7. Para comprobar la firma, ejecute el siguiente comando:

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

Verificación del archivo AWS SCT MSI en Microsoft Windows

AWS proporciona otro nivel de validación además de la suma de comprobación del archivo de distribución. El archivo MSI tiene una firma digital que puede comprobar para asegurarse de que está firmado. AWS

Para comprobar el archivo AWS SCT MSI en Windows

1. Descargue el archivo AWS SCT de distribución mediante los enlaces de la sección de instalación.
2. Verificar la suma de comprobación del archivo de AWS SCT distribución.
3. Extraiga el contenido del archivo de distribución. Localice el archivo MSI que desea verificar.
4. En Windows Explorer, haga clic con el botón derecho del ratón en el archivo MSI y seleccione Properties.
5. Elija la pestaña Firmas digitales.
6. Verifique que la firma digital es de Amazon Services LLC.

Descarga de los controladores de base de datos necesarios

AWS SCT Para que funcione correctamente, descargue los controladores JDBC para los motores de bases de datos de origen y destino. Si utiliza una plataforma de base de datos de destino virtual, no necesita descargar el controlador JDBC para el motor de base de datos de destino. Para obtener más información, consulte Usar destinos virtuales.

Después de descargar los controladores, aportará la ubicación de los archivos de los controladores. Para obtener más información, consulte Almacenamiento de rutas a los controladores en la configuración global.

Puede descargar los controladores de la base de datos de las siguientes ubicaciones.

> **Important**
>
> Descargue la última versión del controlador disponible. La siguiente tabla incluye la versión más baja del controlador de base de datos compatible con. AWS SCT

Motor de base de datos	Controladores	Ubicación de descarga
Amazon Aurora MySQL-Compatible Edition	`mysql-connector-java-5.1.6.jar`	https://www.mysql.com/products/connector/
Edición de Amazon Aurora compatible con PostgreSQL	`postgresql-42.2.19.jar`	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
Amazon EMR	`HiveJDBC42.jar`	http://awssupportdatasvcs.com/bootstrap-actions/Simba/latest/
Amazon Redshift	`redshift-jdbc42-2.1.0.9.jar`	https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
Amazon Redshift sin servidor	`redshift-jdbc42-2.1.0.9.jar`	https://s3.amazonaws.com/redshift-downloads/drivers/jdbc/2.1.0.9/redshift-jdbc42-2.1.0.9.zip
Apache Hive	`hive-jdbc-2.3.4-standalone.jar`	https://repo1.maven.org/maven2/org/apache/hive/hive-jdbc/2.3.4/hive-jdbc-2.3.4-standalone.jar

Motor de base de datos	Controladores	Ubicación de descarga
Base de datos SQL Azure	`mssql-jdbc-7.2.2.jre11.jar`	https://docs.microsoft.com/en-us/sql/connect/jdbc/ release-notes-for-the -jdbc-driver? sql-server-verview= 15 #72
Azure Synapse Analytics	`mssql-jdbc-7.2.2.jre11.jar`	https://docs.microsoft.com/en-us/sql/connect/jdbc/ release-notes-for-the -jdbc-driver? sql-server-verview= 15 #72
Base de datos Greenplum	`postgresql-42.2.19.jar`	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
IBM Db2 para z/OS	`db2jcc-db2jcc4.jar`	https://www.ibm.com/support/pages/db2 - - downloads-db2-zos jdbc-driver-versions-and
IBM Db2 LUW	`db2jcc-db2jcc4.jar`	https://www.ibm.com/support/pages/node/382667
MariaDB	`mariadb-java-client-2.4.1.jar`	https://downloads.mariadb.com/Connectors/java/connector-java-2.4.1/ mariadb-java-client -2.4.1.jar
Microsoft SQL Server	`mssql-jdbc-10.2.jar`	https://docs.microsoft.com/en-us/sql/connect/jdbc/ -? download-microsoft-jdbc-driver for-sql-server vista = 15 sql-server-ver
MySQL	`mysql-connector-java-8.0.15.jar`	https://dev.mysql.com/downloads/connector/j/

Motor de base de datos	Controladores	Ubicación de descarga
Netezza	`nzjdbc.jar` Utilice el software de las herramientas del cliente. Descargue la versión del controlador 7.2.1, que es compatible con la versión de almacenamiento de datos 7.2.0.	http://www.ibm.com/support/knowledgecenter/SSULQD_7.2.1/com.ibm.nz.datacon.doc/c_datacon_plg_overview.html
Oracle	`ojdbc8.jar` Las versiones del controlador 8 y posteriores son compatibles.	https://www.oracle.com/database/technologies/jdbc-ucp-122-downloads.html
PostgreSQL	`postgresql-42.2.19.jar`	https://jdbc.postgresql.org/download/postgresql-42.2.19.jar
SAP ASE (Sybase ASE)	`jconn4.jar`	https://dcx.sap.com/sqla170/en/html/3bd588c76c5f10149824867aa281fadd.html
Snowflake	`snowflake-jdbc-3.9.2.jar` Para obtener más información, consulte Descargar / Integrar el controlador JDBC.	https://repo1.maven.org/maven2/net/snowflake/snowflake-jdbc/3.9.2/snowflake-jdbc-3.9.2.jar

Motor de base de datos	Controladores	Ubicación de descarga

Teradata	`terajdbc4.jar`
	
	`tdgssconfig.jar`
	
	Para la versión 16.20.00.11 y superior del controlador JDBC de Teradata, no necesita el archivo `tdgssconfig.jar` .
		https://downloads.teradata.com/download/connectivity/jdbc-driver

Vertica	`vertica-jdbc-9.1.1-0.jar` Las versiones del controlador 7.2.0 y posteriores son compatibles.	https://www.vertica.com/client_drivers/9.1.x/9.1.1-0/vertica-jdbc-9.1.1-0.jar

Instala controladores JDBC en Linux

Puede seguir los siguientes pasos para instalar los controladores JDBC en su sistema Linux y utilizarlos con. AWS SCT

Para instalar los controladores de JDBC en su sistema Linux

1. Cree un directorio para almacenar los controladores de JDBC.

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

2. Instale el controlador de JDBC para el motor de su base de datos con los comandos que se muestran a continuación.

Motor de base de datos	Comandos de instalación
Amazon Aurora (compatible con MySQL)	`PROMPT> cd /usr/local/jdbc-drivers` `PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz`
Amazon Aurora (compatible con PostgreSQL)	`PROMPT> cd /usr/local/jdbc-drivers` `PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .`
Microsoft SQL Server	`PROMPT> cd /usr/local/jdbc-drivers` `PROMPT> sudo tar xzvf /tmp/sqljdbc_X.X.X_enu.tar.gz`
MySQL	`PROMPT> cd /usr/local/jdbc-drivers` `PROMPT> sudo tar xzvf /tmp/mysql-connector-java-X.X.X.tar.gz`

Oracle	`PROMPT> cd /usr/local/jdbc-drivers`
	`PROMPT> sudo mkdir oracle-jdbc`
	`PROMPT> cd oracle-jdbc`
	`PROMPT> sudo cp  -a /tmp/ojdbc8.jar .`

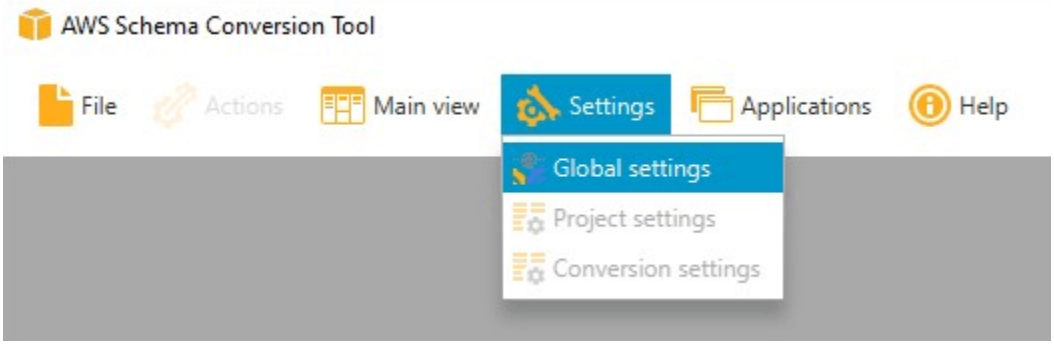
PostgreSQL	`PROMPT> cd /usr/local/jdbc-drivers` `PROMPT> sudo cp -a /tmp/postgresql-X.X.X.jre7.tar .`

Almacenamiento de rutas a los controladores en la configuración global

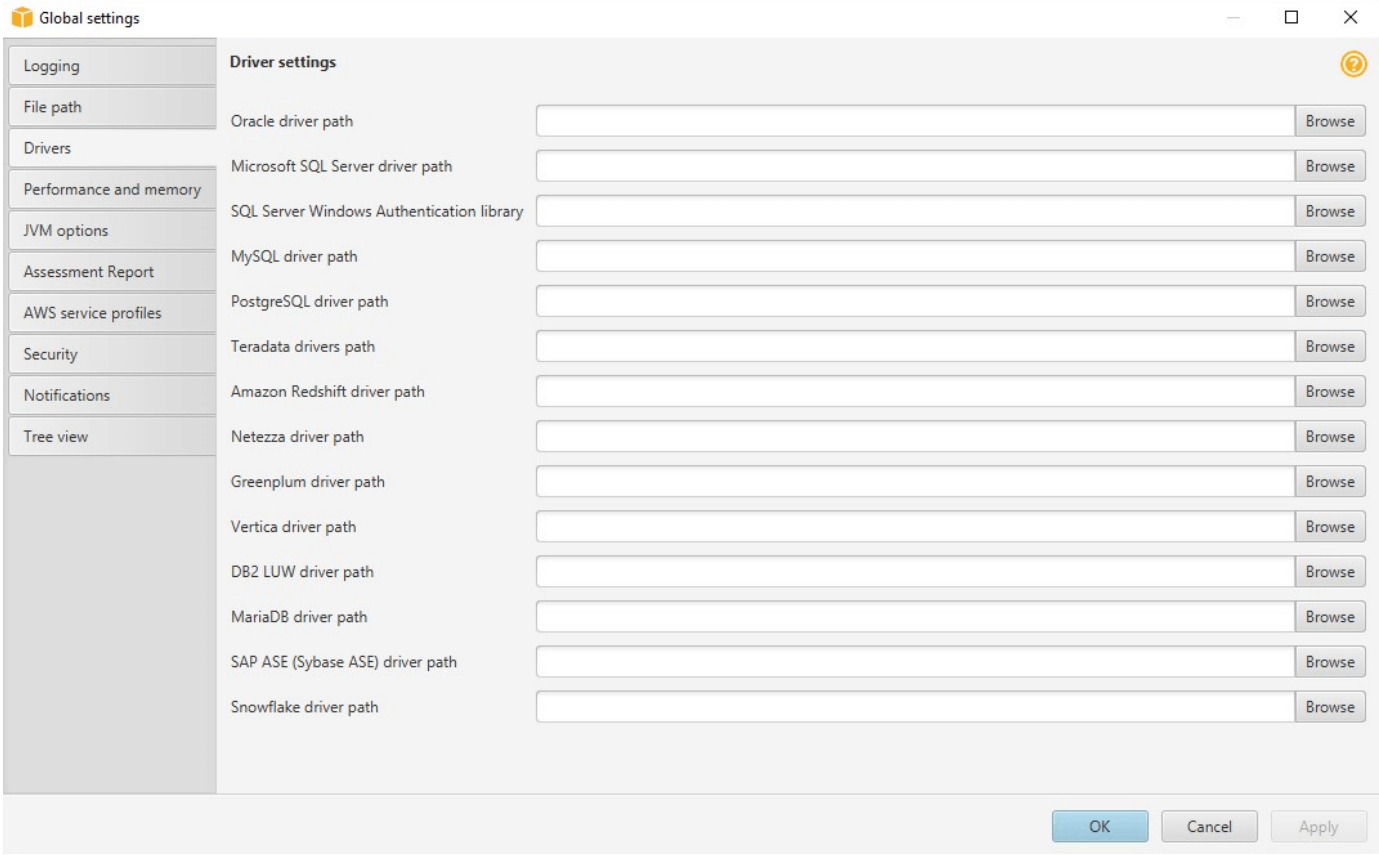
Una vez que haya descargado e instalado los controladores JDBC necesarios, puede establecer la ubicación global de los controladores en la configuración. AWS SCT Si no configura la ubicación de los controladores globalmente, la aplicación le pedirá la ubicación de los controladores al conectarse a una base de datos.

Para actualizar las ubicaciones de los archivos de los controladores

1. En AWS SCT, elija Configuración y, a continuación, Configuración global.



2. En Global settings, seleccione Drivers. Agregue la ruta al archivo del controlador JDBC para su motor de base de datos de origen y su motor de base de datos de la instancia de base de datos de Amazon RDS de destino.



3. Cuando haya terminado de agregar las rutas de los controladores, seleccione OK.

Actualizando AWS SCT

AWS se actualiza periódicamente AWS SCT con nuevas características y funcionalidades. Si está actualizando desde una versión anterior, cree un AWS SCT proyecto nuevo y vuelva a convertir los objetos de la base de datos que esté utilizando.

Puede comprobar si existen actualizaciones para AWS SCT.

Para comprobar si hay actualizaciones de AWS SCT

1. Cuando estés dentro AWS SCT, selecciona Ayuda y, a continuación, selecciona Buscar actualizaciones.
2. En el cuadro de diálogo Check for Updates, elija What's New. Si el enlace no aparece, tiene la versión más reciente.

AWS SCT CLI

Puede descargar la AWS SCT CLI para usarla en la línea de comandos. Para descargar el JAR, utilice el siguiente enlace:

AWSSchemaConversionToolBatch.jar

Utilización de la interfaz de usuario de AWS SCT

Utilice los siguientes temas para ayudarle a trabajar con la interfaz de usuario de AWS SCT. Para obtener información sobre la instalación de AWS SCT, consulte Instalación, verificación y actualización AWS SCT.

Temas

- La ventana de proyecto de AWS SCT
- Inicio de AWS SCT
- Creación de un proyecto de AWS SCT
- Uso de un asistente de proyecto nuevo en AWS SCT
- Cómo guardar y abrir un proyecto de AWS SCT
- Cómo agregar servidores de bases de datos a un proyecto de AWS SCT
- Ejecutar AWS SCT en el modo sin conexión
- Utilización de los filtros de árbol de AWS SCT
- Cómo ocultar esquemas en la vista de árbol de AWS SCT
- Creación y revisión del informe de evaluación de la migración de la base de datos
- Conversión de un esquema
- Aplicación del esquema convertido a su instancia de base de datos de destino
- Almacenamiento de perfiles de servicio de AWS en AWS SCT
- Utilizar AWS Secrets Manager
- Almacenar contraseñas de la base de datos
- Utilizar la vista UNION ALL en proyectos con tablas con particiones
- Métodos abreviados de teclado para AWS SCT

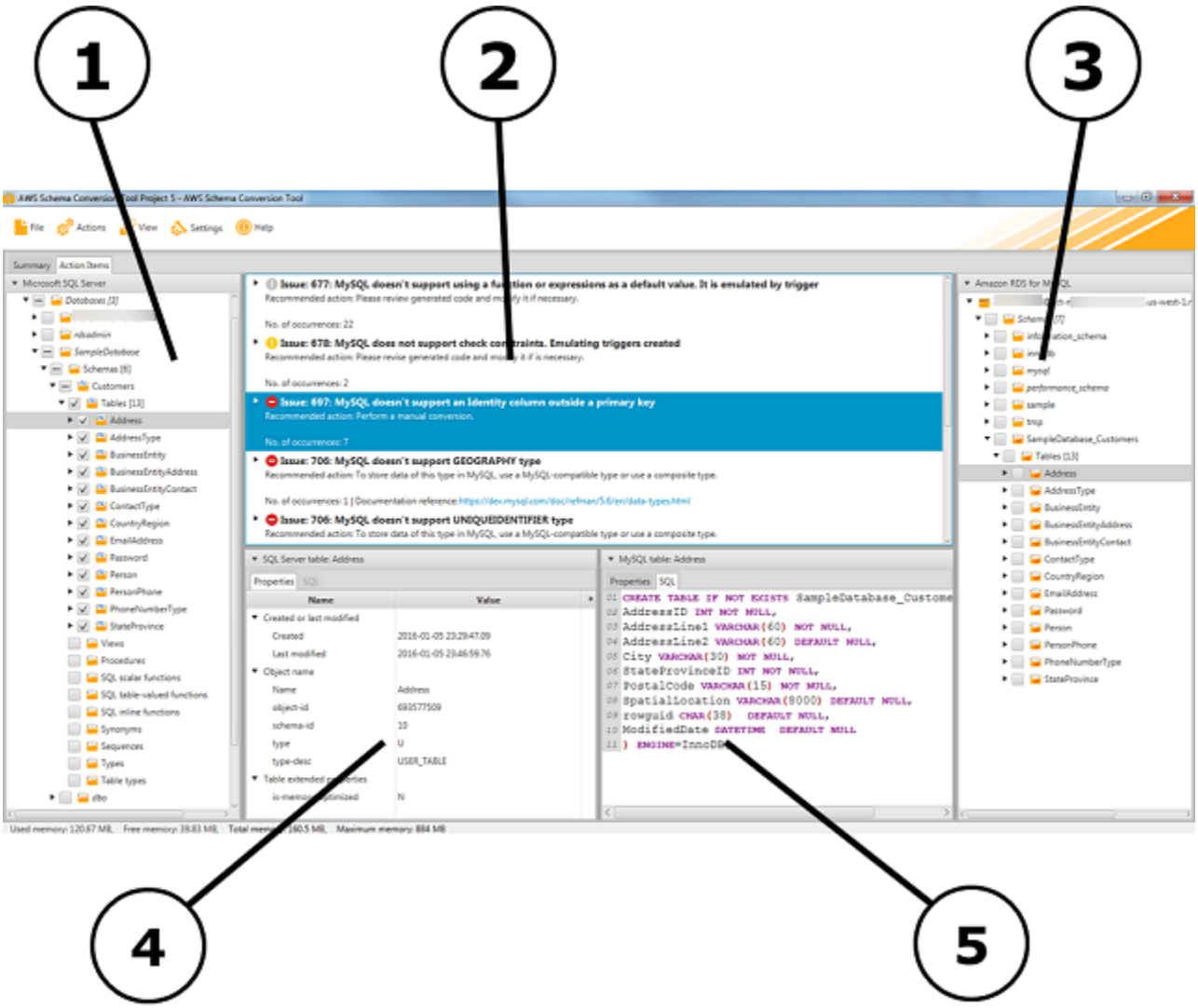
La ventana de proyecto de AWS SCT

A continuación se ilustra lo que se puede ver en AWS SCT al crear un proyecto de migración de esquema y convertir un esquema posteriormente.

1. En el panel izquierdo, el esquema de la base de datos de origen se presenta en una vista en árbol. Su esquema de base de datos se "carga en diferido". Dicho de otra manera, al seleccionar

un elemento de la vista en árbol, AWS SCT obtiene y muestra el esquema actual de su base de datos de origen.

2. En el panel superior central, aparecerán los elementos de acción para los elementos del esquema del motor de la base de datos de origen que no se hayan podido convertir automáticamente al motor de la base de datos de destino.
3. En el panel derecho, el esquema de su instancia de base de datos de destino se presenta en una vista en árbol. Su esquema de base de datos se "carga en diferido". Es decir, en el momento en que selecciona un elemento de la vista en árbol, AWS SCT obtiene y muestra el esquema actual de su base de datos de destino.



4. En el panel inferior izquierdo, al elegir un elemento de esquema, se muestran las propiedades. Estas describen el elemento de esquema de origen y el comando SQL para crear ese elemento en la base de datos de origen.
5. En el panel inferior derecho, al elegir un elemento de esquema, se muestran las propiedades. Estas describen el elemento de esquema de origen y el comando SQL para crear ese elemento en la base de datos de destino. Puede editar este comando SQL y guardar el comando actualizado con su proyecto.

Inicio de AWS SCT

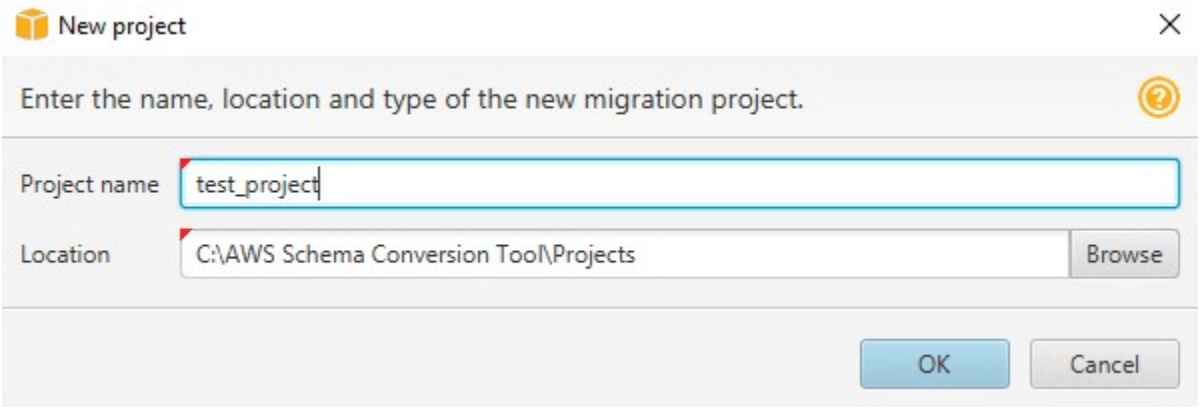
Para iniciar AWS Schema Conversion Tool, haga doble clic en el icono de la aplicación.

Creación de un proyecto de AWS SCT

Utilice el siguiente procedimiento para crear un proyecto de AWS Schema Conversion Tool.

Para crear su proyecto

1. Inicie AWS Schema Conversion Tool.
2. En el menú Archivo, seleccione Proyecto nuevo. Aparece el cuadro de diálogo Proyecto nuevo.



3. Introduzca un nombre para su proyecto, que se almacenará localmente en su equipo.
4. Introduzca la ubicación del archivo local de su proyecto.
5. Elija OK (Aceptar) para crear su proyecto de AWS SCT.
6. Elija Agregar origen para agregar una nueva base de datos de origen al proyecto de AWS SCT. Puede agregar varias bases de datos de origen al proyecto de AWS SCT.
7. Seleccione Agregar destino para agregar una plataforma de destino nueva al proyecto de AWS SCT. Puede agregar varias plataformas de destino al proyecto de AWS SCT.

8. Elija el esquema de la base de datos de origen en el panel izquierdo.
9. En el panel derecho, especifique la plataforma de la base de datos de destino para el esquema de origen seleccionado.
10. Seleccione Crear asignación. Este botón se activa después de elegir el esquema de la base de datos de origen y la plataforma de la base de datos de destino. Para obtener más información, consulte Creación de reglas de asignación.

Su proyecto de AWS SCT ya está configurado. Puede guardar su proyecto, crear un informe de evaluación de la migración de la base de datos y convertir los esquemas de la base de datos de origen.

Uso de un asistente de proyecto nuevo en AWS SCT

Puede crear un proyecto de migración de bases de datos nuevo utilizando el asistente de proyecto nuevo. Este asistente le ayuda a determinar el destino de la migración y a conectarse a sus bases de datos. Estima la complejidad de una migración para todos los destinos admitidos. Tras ejecutar el asistente, AWS SCT genera un informe resumido para la migración de la base de datos a distintos destinos. Puede utilizar este informe para comparar los posibles destinos y elegir la ruta de migración óptima.

Para ejecutar el asistente de proyecto nuevo

1. Elija su base de datos de origen.

   a. Inicie AWS Schema Conversion Tool.

   b. En el menú Archivo, seleccione Asistente de proyecto nuevo. Se abre el cuadro de diálogo Crear un proyecto de migración de bases de datos nuevo.

   c. Para introducir la información de conexión a la base de datos de origen, siga las instrucciones siguientes:

Parámetro	Acción
Project name	Introduzca un nombre para su proyecto, que se almacenará localmente en su equipo.
Location	Introduzca la ubicación del archivo local de su proyecto.

Parámetro	Acción
Source type	Elija una de las siguientes opciones: Base de datos de SQL, Base de datos de NoSQL o ETL. Si desea ver el informe resumido que incluye todos los destinos de migración, elija Base de datos de SQL.
Source engine	Elija el motor de la base de datos de origen.
Migration strategy	Elija una de las siguientes opciones: • Quiero cambiar de motor y optimizarlo para la nube. Esta opción convierte la base de datos de origen en un nuevo motor de base de datos. • Quiero mantener el mismo motor pero optimizarlo para la nube: esta opción mantiene el motor de base de datos tal como está y mueve la base de datos de las instalaciones a la nube. • Quiero ver un informe combinado sobre el cambio del motor de base de datos y la optimización para la nube. Esta opción compara la complejidad de la migración de todas las opciones de migración disponibles. Si desea ver el informe de evaluación agregado que incluye todos los destinos de migración, elija la última opción.

d. Elija Siguiente. Se abre la página Conectar a la base de datos de origen.

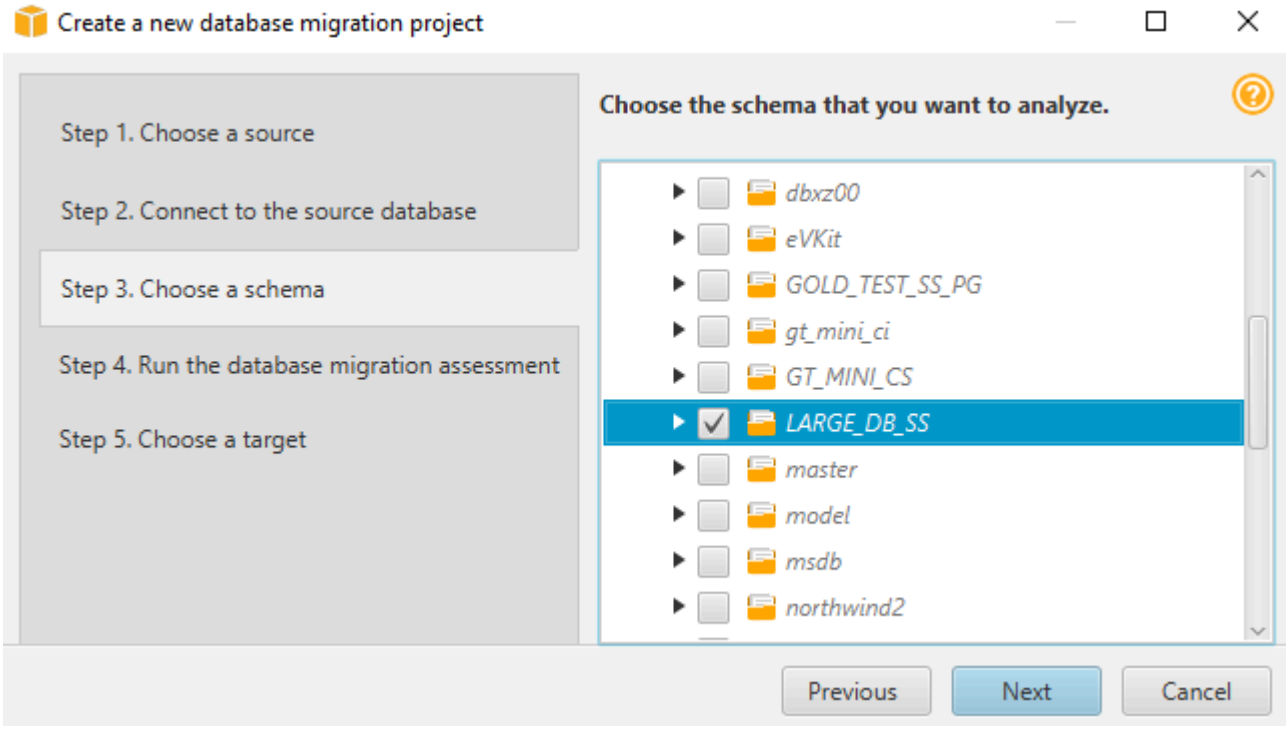
2. Conéctese a su base de datos de origen.

   a. Proporcione la información de conexión para la base de datos de origen. Los parámetros de conexión dependen del motor de base de datos de origen. Asegúrese de que el usuario que utiliza para el análisis de la base de datos de origen tiene los permisos correspondientes. Para obtener más información, consulte Orígenes para AWS SCT.

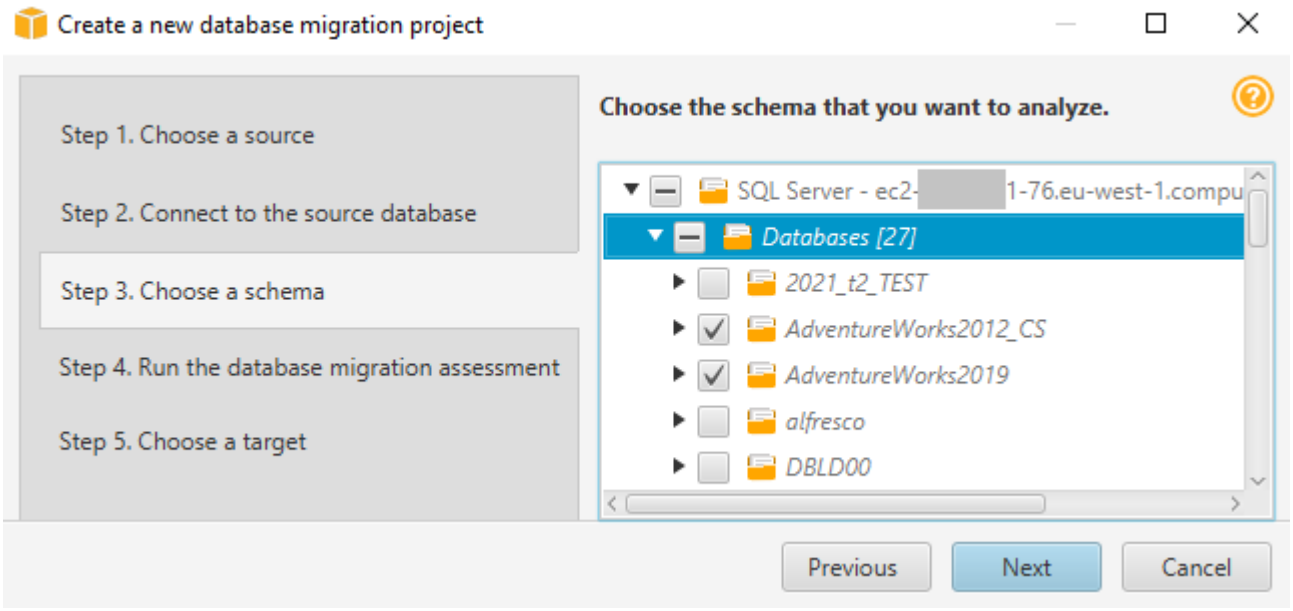
   b. Elija Siguiente. Se abre la página Elegir un esquema.

3. Seleccione el esquema de base de datos.

   a. Seleccione la casilla de verificación del nombre de los esquemas que desea evaluar y elija el esquema en sí. El nombre del esquema se resalta en azul cuando se selecciona y el botón Siguiente está disponible.



b. Si desea evaluar varios esquemas de base de datos, active las casillas de verificación de todos los esquemas y, a continuación, elija el nodo principal. Para que la evaluación se realice correctamente, debe elegir el nodo principal. Por ejemplo, para una base de datos de SQL Server de origen, elija el nodo Bases de datos. El nombre del nodo principal se resalta en azul cuando se selecciona y el botón Siguiente está disponible.



c. Seleccione Siguiente. AWS SCT analiza los esquemas de base de datos de origen y crea un informe de evaluación de la migración de la base de datos. La cantidad de objetos de base

de datos presentes en los esquemas de base de datos de origen afecta al tiempo que se tarda en ejecutar la evaluación. Al finalizar, se abrirá la página Ejecutar la evaluación de la migración de la base de datos.

4. Ejecute la evaluación de la migración de la base de datos

   a. Puede revisar y comparar los informes de evaluación de los distintos destinos de migración o guardar una copia local de los archivos del informe de evaluación para analizarlos más a fondo.

   b. Guarde una copia local del informe de evaluación de la migración de la base de datos. Seleccione Guardar, introduzca la ruta a la carpeta para guardar los archivos y, a continuación, seleccione Guardar. AWS SCT guarda los archivos del informe de evaluación en la carpeta especificada.

   c. Elija Siguiente. Se abre la página Elegir un destino.

5. Elija su base de datos de destino.

   a. En Motor de destino, elija el motor de base de datos de destino que decida utilizar en función del informe de evaluación.

   b. Proporciones la información de conexión para la base de datos de destino. Los parámetros de conexión que ve dependen del motor de base de datos de destino seleccionado. Asegúrese de que el usuario especificado para la base de datos de destino tiene los permisos necesarios. Para obtener más información sobre los permisos necesarios, consulte las secciones en las que se describen los permisos para las bases de datos de destino en Orígenes para AWS SCT y Permisos para Amazon Redshift como destino.

   c. Seleccione Finalizar. AWS SCT crea el proyecto y agrega las reglas de asignación. Para obtener más información, consulte Creación de reglas de asignación.

Ahora puede usar el proyecto de AWS SCT para convertir los objetos de la base de datos de origen.

Cómo guardar y abrir un proyecto de AWS SCT

Utilice el siguiente procedimiento para guardar un proyecto de AWS Schema Conversion Tool.

Para guardar el proyecto

1. Inicie AWS Schema Conversion Tool.
2. En el menú Archivo, seleccione Proyecto nuevo.

AWS SCT guarda el proyecto en la carpeta que especificó al crear el proyecto.

Utilice el siguiente procedimiento para abrir un proyecto de AWS Schema Conversion Tool.

Para abrir su proyecto

1. En el menú Archivo, elija Abrir proyecto. A continuación, aparece un cuadro de diálogo Abrir.
2. Elija la carpeta del proyecto y, a continuación, elija el archivo Windows Script Component (*.sct).
3. AWS SCT abre su proyecto, pero no se conecta automáticamente a las bases de datos de origen y de destino. Elija Conectar al servidor en la parte superior de los árboles de esquemas de base de datos para conectarse a las bases de datos de origen y de destino.

Si abre un proyecto guardado en la versión 1.0.655 o anterior de AWS SCT, AWS SCT crea automáticamente reglas de asignación para todos los esquemas de la base de datos de origen a la plataforma de base de datos de destino. Para agregar otras plataformas de bases de datos de destino, elimine las reglas de asignación existentes y, a continuación, cree reglas de asignación nuevas. Para obtener más información sobre la creación de reglas de asignación, consulte Creación de reglas de asignación.

Cómo agregar servidores de bases de datos a un proyecto de AWS SCT

Puede agregar varios servidores de bases de datos de origen y destino a un proyecto de AWS Schema Conversion Tool.

Para agregar un servidor al proyecto

1. Inicie AWS Schema Conversion Tool.
2. Cree un proyecto nuevo o abra un proyecto existente.
3. Seleccione Agregar origen en el menú para añadir una base de datos de origen nueva.
4. Elija una plataforma de base de datos y especifique las credenciales de conexión a la base de datos. Para obtener más información acerca de la conexión a una base de datos de origen, consulte Orígenes para AWS SCT.

Utilice el siguiente procedimiento para conectarse a su base de datos de origen.

Para conectarse a su base de datos

1. Abra el menú contextual (clic secundario) de un servidor de base de datos y elija Establecer conexión.

   También puede elegir Conectar al servidor en la parte superior del árbol de esquemas de base de datos.
2. Escriba la contraseña para conectarse al servidor de base de datos de origen.
3. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
4. Seleccione Conectar para conectarse a su base de datos de origen.

Utilice el siguiente procedimiento para eliminar un servidor de base de datos del proyecto de AWS SCT.

Para eliminar un servidor de base de datos

1. Elija el servidor de base de datos que desee eliminar.
2. Abra el menú contextual (clic secundario) y, a continuación, elija Eliminar del proyecto.

   AWS SCT elimina el servidor de base de datos seleccionado, todas las reglas de asignación, los resultados de conversión y otros metadatos relacionados con este servidor.

Ejecutar AWS SCT en el modo sin conexión

Puede ejecutar AWS Schema Conversion Tool en modo sin conexión. A continuación, puede aprender a trabajar con un proyecto de AWS SCT existente sin conexión con la base de datos de origen.

AWS SCT no requiere una conexión a la base de datos de origen para ejecutar las siguientes operaciones:

- Agregar reglas de asignación.
- Crear informes de evaluación de la migración de la base de datos.
- Convertir los esquemas y el código de base de datos.
- Editar el código fuente y el código convertido.
- Guardar el código fuente y convertido como scripts SQL en un archivo de texto.

Antes de usar AWS SCT en modo sin conexión, conéctese a la base de datos de origen, cargue los metadatos y guarde el proyecto. Abra este proyecto o desconéctese del servidor de la base de datos de origen para usar AWS SCT en modo sin conexión.

Para ejecutar AWS SCT en modo sin conexión

1. Inicie AWS Schema Conversion Tool y cree un proyecto nuevo. Para obtener más información, consulte Creación de un proyecto de AWS SCT.
2. Agregue un servidor de base de datos de origen y conéctese a la base de datos de origen. Para obtener más información, consulte Cómo agregar servidores de bases de datos a un proyecto de AWS SCT.
3. Agregue un servidor de base de datos de destino o utilice una plataforma de base de datos de destino virtual. Para obtener más información, consulte Usar destinos virtuales.
4. Cree una regla de asignación para definir la plataforma de base de datos de destino para la base de datos de origen. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT.
5. Haga clic en Ver y, a continuación, seleccione Vista principal.
6. En el panel de la izquierda que muestra los objetos de su base de datos de origen, seleccione esquemas de base de datos de origen. Abra el menú contextual (clic secundario) para el objeto y, a continuación, seleccione Cargar esquema. Esta operación carga todos los metadatos del esquema de origen en el proyecto de AWS SCT.

   Las operaciones Crear informe y Convertir esquema también cargan todos los metadatos del esquema de origen en el proyecto de AWS SCT. Si ejecutó una de estas operaciones desde el menú contextual, omita la operación Cargar esquema.
7. En el menú Archivo, seleccione Guardar proyecto para guardar los metadatos de la base de datos de origen en el proyecto.
8. Seleccione Desconectar del servidor para desconectarse de la base de datos de origen. Ahora puede usar AWS SCT en el modo sin conexión.

Utilización de los filtros de árbol de AWS SCT

Para migrar datos de un origen a un destino, AWS SCT carga todos los metadatos de las bases de datos de origen y de destino en una estructura de árbol. Esta estructura aparece en la AWS SCT como la vista de árbol en la ventana del proyecto principal.

Algunas bases de datos puede tener un gran número de objetos en la estructura de árbol. Puede utilizar filtros de árbol en la AWS SCT para buscar objetos en las estructuras de árbol de origen y de destino. Cuando utiliza un filtro de árbol, no cambia los objetos que se convierten al convertir la base de datos. El filtro cambia únicamente lo que se ve en el árbol.

Los filtros de árbol funcionan con objetos que la AWS SCT ha precargado. En otras palabras, la AWS SCT no carga objetos de la base de datos durante las búsquedas. Este enfoque significa que la estructura de árbol, por lo general, contiene menos objetos que los que están presentes en la base de datos.

Cuando utilice filtros de árbol, tenga en cuenta lo siguiente:

- El valor predeterminado del filtro es ANY, lo que significa que el filtro utiliza una búsqueda de nombre para encontrar objetos.
- Al seleccionar uno o varios tipos de objeto, verá solo esos tipos de objetos en el árbol.
- Puede utilizar el filtro enmascarar para mostrar diferentes tipos de símbolos, incluidos Unicode, espacios y caracteres especiales. El carácter “%” sirve de comodín para cualquier símbolo.
- Después de aplicar un filtro, el recuento solo muestra el número de objetos filtrados.

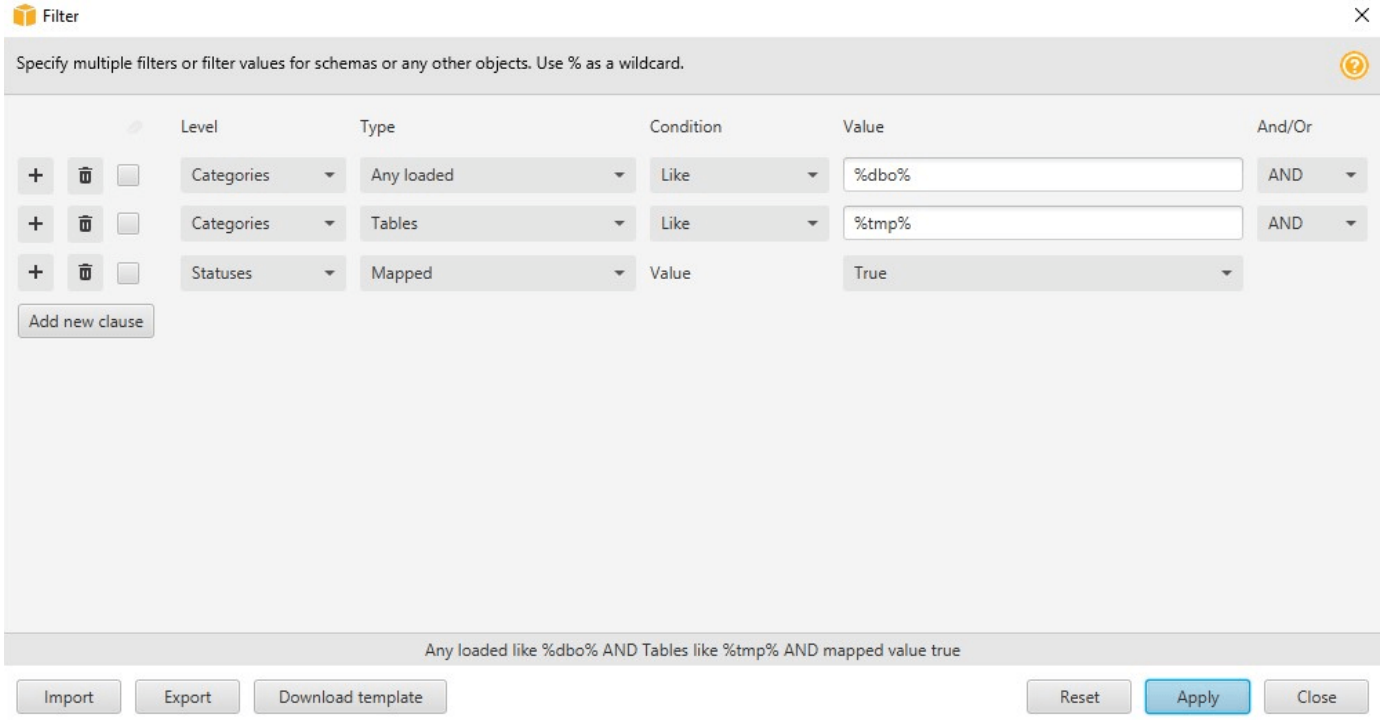
Para crear un filtro de árbol

1. Abrir un proyecto de AWS SCT existente.
2. Conéctese a la base de datos a la que desea aplicar el filtro de árbol.
3. Elija el icono de filtro.

   El icono de deshacer filtro está difuminado porque no hay ningún filtro aplicado en la actualidad.
4. Introduzca la siguiente información en el cuadro de diálogo Filtro. Las opciones en el cuadro de diálogo son diferentes para cada motor de base de datos.

Opciones de filtro de AWS SCT	Acción
Nivel	Elija Categorías para filtrar los objetos por categorías.

Opciones de filtro de AWS SCT	Acción
	Elija Estados para filtrar los objetos por estados.
Tipo	En Categorías, en Nivel, elija las categorías de los objetos filtrados. Seleccione Cualquiera cargado para mostrar los objetos de todas las categorías. En Estados, en Nivel, elija el estado de los objetos filtrados. Puede elegir una de las siguientes opciones: • Convertido para mostrar todos los objetos convertidos • Con acciones para mostrar todos los objetos que tienen problemas de conversión • Cifrado para mostrar todos los objetos cifrados
Condición	En Categorías, en Nivel, elija la condición de filtrado entre Me gusta y No me gusta. En Estados, en Nivel, la opción de condición de filtrado no está disponible.
Valor	En Categorías, en Nivel, introduzca el valor para filtrar el árbol por este valor. Utilice el porcentaje (%) como comodín para mostrar todos los objetos. En Estados, en Nivel, elija el valor entre True y False.
Y/O	Elija en los operadores lógicos `AND` o `OR` para aplicar varias cláusulas de filtro.



5. Seleccione Agregar cláusula nueva para agregar una cláusula de filtro adicional. AWS SCT puede aplicar varias cláusulas de filtro utilizando los operadores `AND` o `OR`.
6. Seleccione Aplicar. Después de seleccionar Apply, se activa el icono de deshacer filtro (al lado del icono de filtro). Utilice este icono si desea eliminar los filtros que aplicó.
7. Seleccione Close para cerrar el cuadro de diálogo.

Al filtrar el esquema que aparece en el árbol, no cambia los objetos que se convierten al convertir su esquema. El filtro cambia únicamente lo que se ve en el árbol.

Importación de una lista de archivos para el filtro de árbol

Puede importar un archivo de valores separados por comas (CSV) con separadores de punto y coma o un archivo JSON que contenga los nombres o valores que desee que utilice el filtro de árbol. Abra un proyecto de AWS SCT existente, conéctese a la base de datos a la que desea aplicar el filtro de árbol y, a continuación, elija el icono de filtro.

Seleccione Descargar plantilla para descargar un ejemplo del archivo. Introduzca el nombre del archivo y seleccione Guardar.

Para descargar la configuración de filtro existente, seleccione Exportar. Introduzca el nombre del archivo y seleccione Guardar.

Para importar una lista de archivos para el filtro de árbol, seleccione Importar. Seleccione un archivo para importar y, luego, seleccione Open. Seleccione Apply y, a continuación, seleccione Close.

Los archivos CSV utilizan punto y coma como separador y tienen el siguiente formato:

- `object_type` es el tipo de objeto que desea encontrar.
- `database_name` es el nombre de la base de datos donde existe este objeto.
- `schema_name` es el nombre del esquema donde existe este objeto.
- `object_name` es el nombre del objeto.
- `import_type` especifica a `include` o `exclude` este elemento del filtro.

Utilice archivos JSON para describir casos de filtrado complejos, como reglas anidadas. Los archivos JSON tienen el siguiente formato:

- `filterGroupType` es el tipo de regla de filtro (operadores lógicos `AND` o `OR`) que se aplica a varias cláusulas de filtro.
- `filterCategory` es el nivel del filtro (Categorías o Estados).
- `names` es la lista de nombres de objetos que se aplica al filtro Categorías.
- `filterCondition` es la condición de filtrado (`LIKE` o `NOT LIKE`) que se aplica al filtro Categorías.
- `transformName` es el nombre de estado que se aplica al filtro Estado.
- `value` es el valor por el que se va a filtrar el árbol.
- `transformValue` es el valor del filtro (`TRUE` o `FALSE`) que se aplica al filtro Estado.

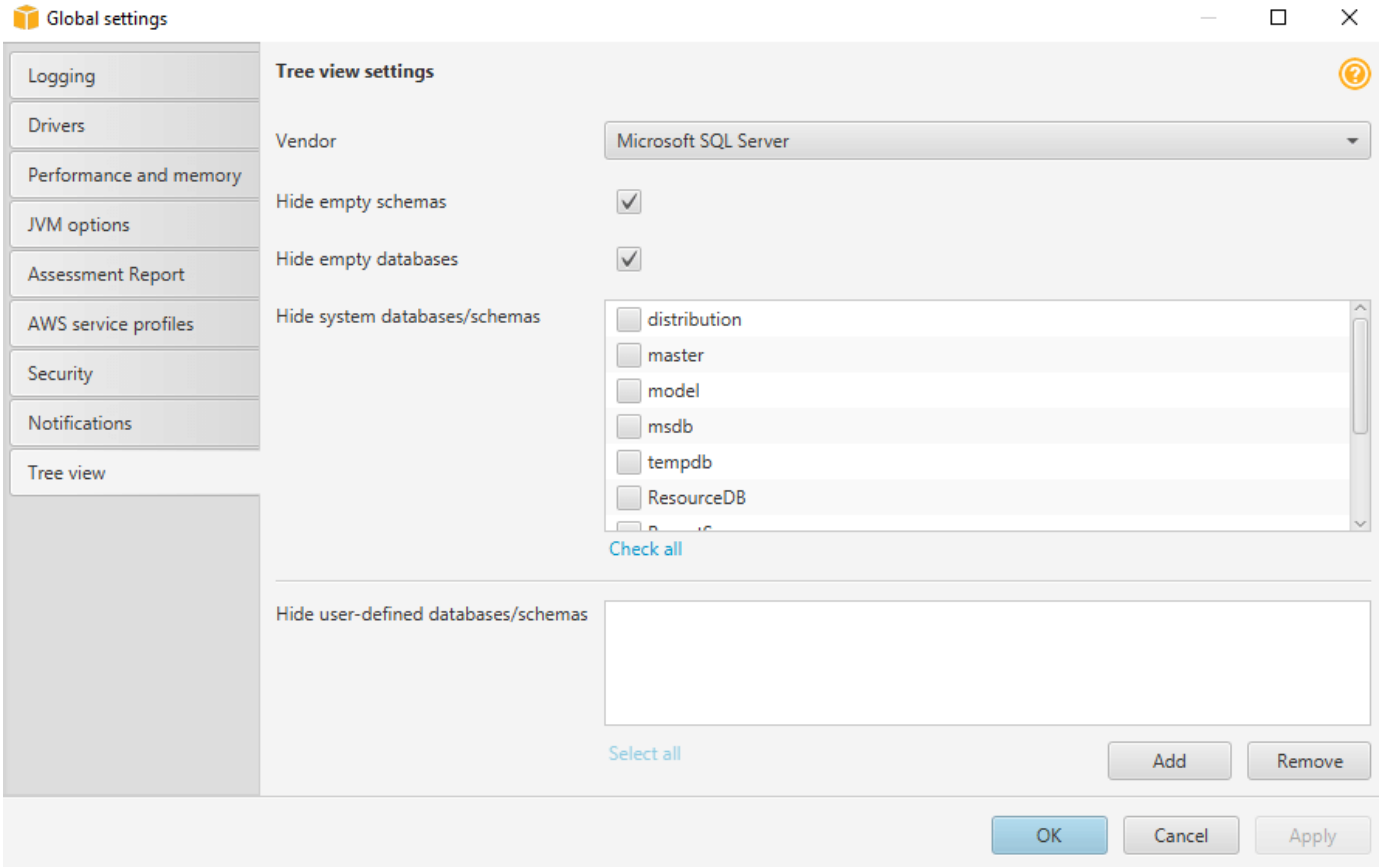
Cómo ocultar esquemas en la vista de árbol de AWS SCT

Al utilizar la configuración de vista de árbol, especifica qué esquemas y bases de datos desea ver en la vista de árbol de la AWS SCT. Puede ocultar esquemas vacíos, bases de datos vacías, bases de datos del sistema y bases de datos y esquemas definidos por el usuario.

Para ocultar bases de datos y esquemas en la vista de árbol

1. Abra un proyecto de AWS SCT.

2. Conéctese al almacén de datos que desea mostrar en la vista de árbol.
3. Seleccione Configuración, Configuración global, Vista de árbol.



4. En la sección Configuración de vista de árbol, siga estos pasos:
   - En Proveedor, elija la plataforma de base de datos.
   - Seleccione Ocultar esquemas vacíos para ocultar los esquemas vacíos de la plataforma de base de datos seleccionada.
   - Seleccione Ocultar esquemas vacíos para ocultar los esquemas vacíos de la plataforma de base de datos seleccionada.
   - En Ocultar bases de datos/esquemas del sistema, seleccione los esquemas y las bases de datos del sistema por nombre para ocultarlos.
   - En Ocultar bases de datos/esquemas definidos por el usuario, escriba los nombres de las bases de datos y esquemas definidos por el usuario que desee ocultar y, a continuación, seleccione Agregar. Los nombres no distinguen entre mayúsculas y minúsculas.
5. Seleccione OK (Aceptar).

Creación y revisión del informe de evaluación de la migración de la base de datos

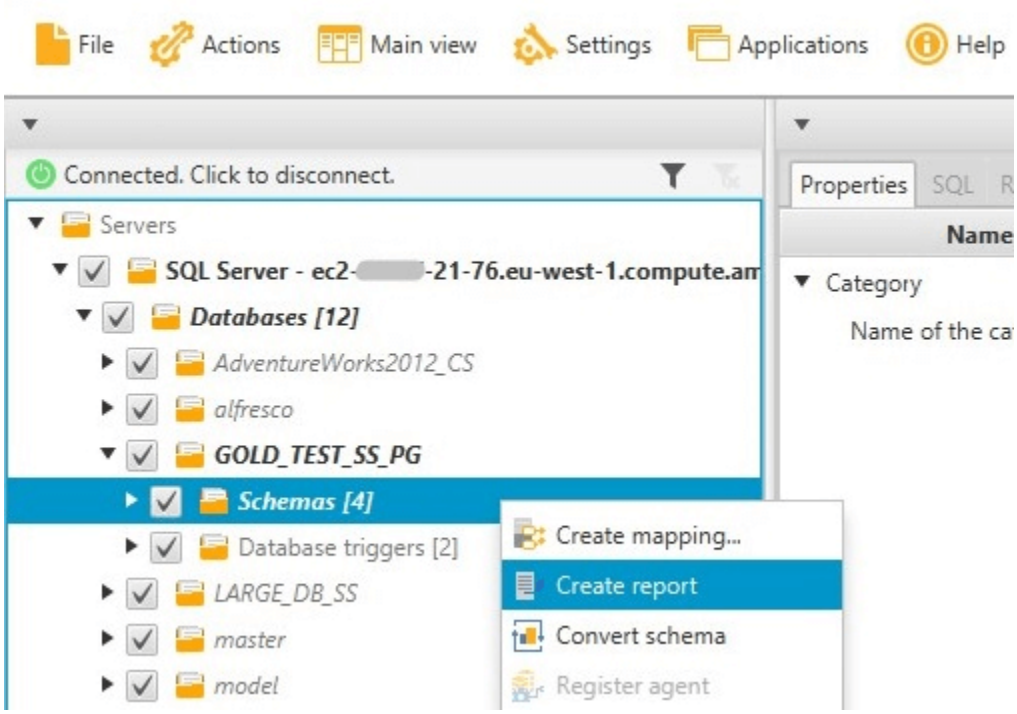
El informe de evaluación de la migración de la base de datos resume todos los elementos de acción de los esquemas que no se pueden convertir automáticamente en el motor de su instancia de Amazon RDS DB de destino. El informe incluye también estimaciones de la cantidad de esfuerzo necesario para escribir el código equivalente de la instancia de base de datos de destino.

Puede crear un informe de evaluación de la migración de la base de datos después de agregar las bases de datos de origen y las plataformas de destino al proyecto y especificar las reglas de asignación.

Para crear y ver el informe de evaluación de la migración de la base de datos

1. Asegúrese de haber creado una regla de asignación para el esquema de la base de datos de origen para crear un informe de evaluación. Para obtener más información, consulte Cómo agregar una regla de asignación nueva.
2. En el menú Ver, seleccione Vista principal.
3. En el panel de la izquierda que muestra el esquema de la base de datos de origen, elija los objetos del esquema para los que desea crear un informe de evaluación.

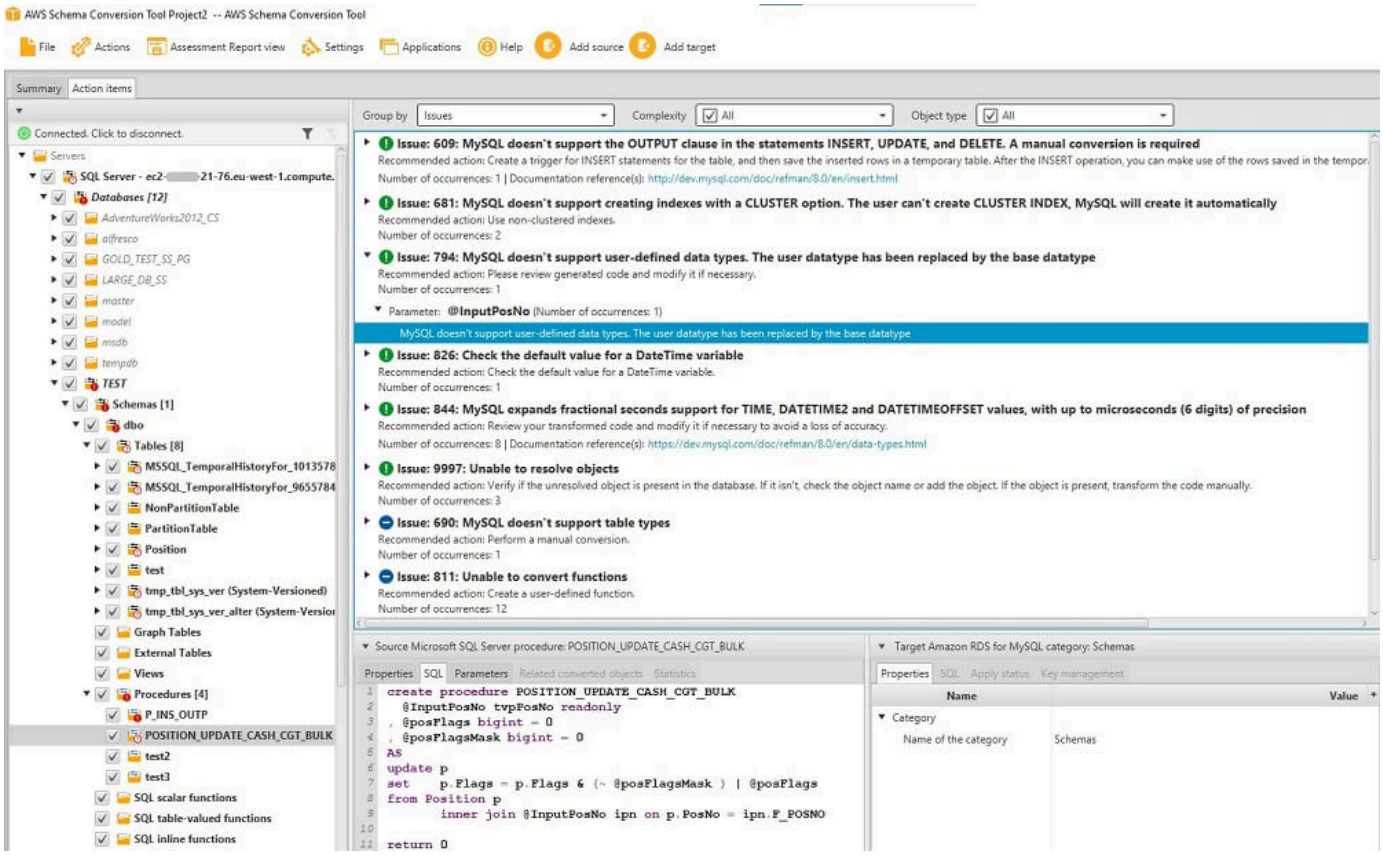
   Asegúrese de haber seleccionado las casillas de verificación de todos los objetos del esquema para los que desee crear un informe de evaluación.
4. Abra el menú contextual (clic secundario) para el objeto y seleccione Create Report.



Se abrirá la vista del informe de evaluación.

5. Haga clic en la pestaña Elementos de acción.

   La pestaña Elementos de acción muestra una lista de elementos que describen el esquema que no puede convertirse automáticamente. Seleccione uno de los elementos de acción de la lista. AWS SCT resalta el elemento de su esquema al que se aplica el elemento de acción, como se muestra a continuación.

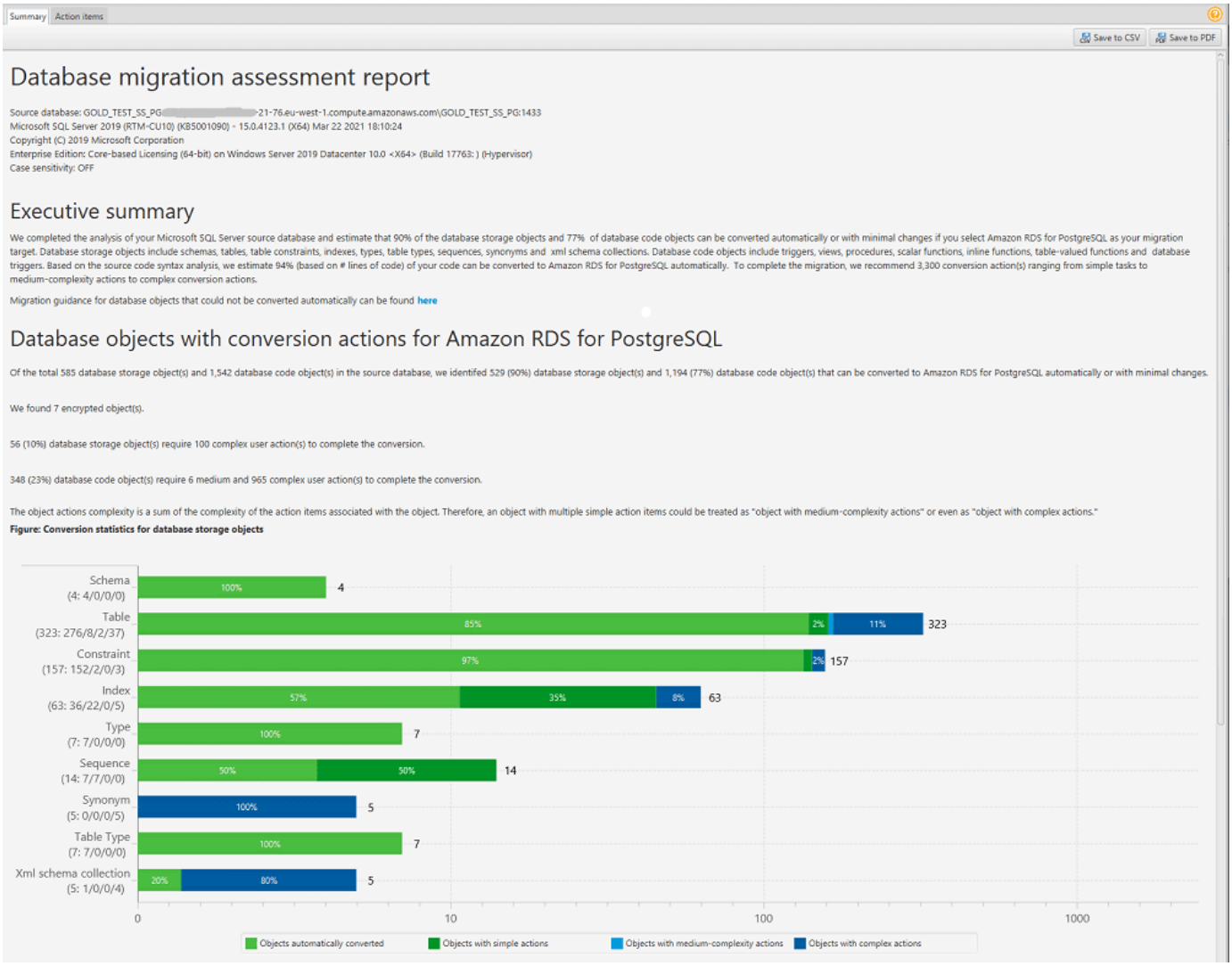


6. Haga clic en la pestaña Summary.

   En la pestaña Summary se muestra la información resumida del informe de evaluación de la migración de la base de datos. Muestra el número de elementos convertidos automáticamente y el número de elementos que no se han convertido automáticamente. El resumen también incluye un cálculo del tiempo que se tardará en crear esquemas en su instancia de base de datos de destino que sean equivalentes a los de la base de datos de origen.

   La sección Evaluación de licencias y soporte en la nube contiene información sobre el traslado de sus esquemas de bases de datos en las instalaciones existentes a una instancia de base de datos de Amazon RDS DB que ejecute el mismo motor. Por ejemplo, si desea cambiar tipos de licencia, esta sección del informe le indica qué características de su base de datos actual deben eliminarse.

   A continuación se muestra un ejemplo de un resumen de informe de evaluación.



7. Seleccione la pestaña Summary y, después, seleccione Save to PDF. El informe de evaluación de la migración de la base de datos se guarda como archivo PDF. El archivo PDF contiene tanto el resumen como información de elementos de acción.

   También puede elegir Guardar en CSV para guardar el informe como un archivo CSV. Al elegir esta opción, AWS SCT crea tres archivos CSV. Estos archivos contienen la información siguiente:

   - Una lista de las acciones de conversión con las acciones recomendadas.
   - Un resumen de los elementos de acción de conversión con una estimación del esfuerzo necesario para convertir una aparición del elemento de acción.
   - Un resumen ejecutivo con una serie de elementos de acción clasificados según el tiempo estimado de conversión.

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 identifed 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.

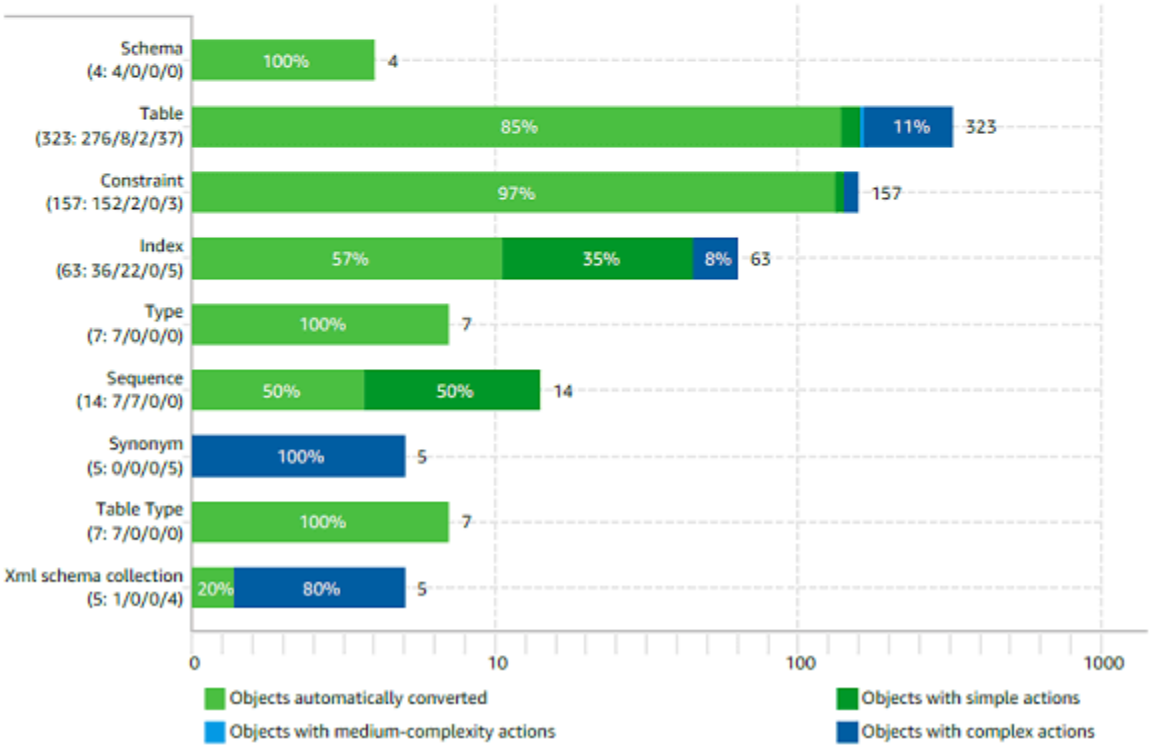
We found 7 encrypted object(s).

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

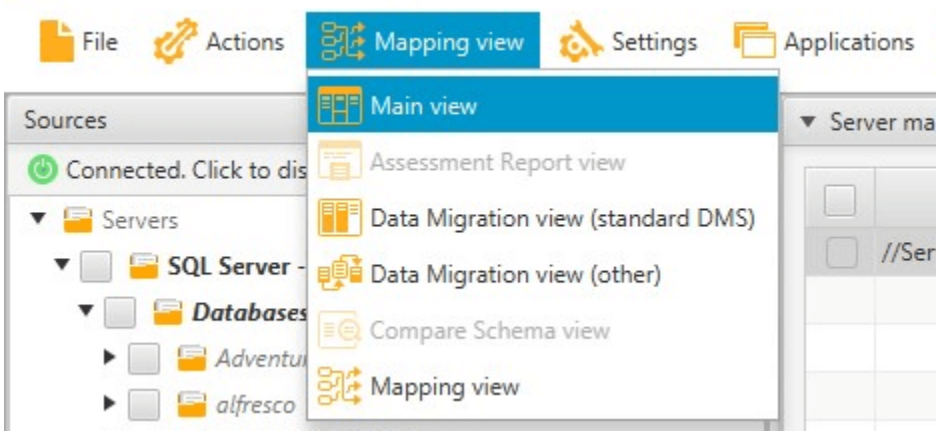


Conversión de un esquema

Después de agregar las bases de datos de origen y destino al proyecto y de crear las reglas de asignación, puede convertir los esquemas de la base de datos de origen. Utilice el siguiente procedimiento para convertir un esquema.

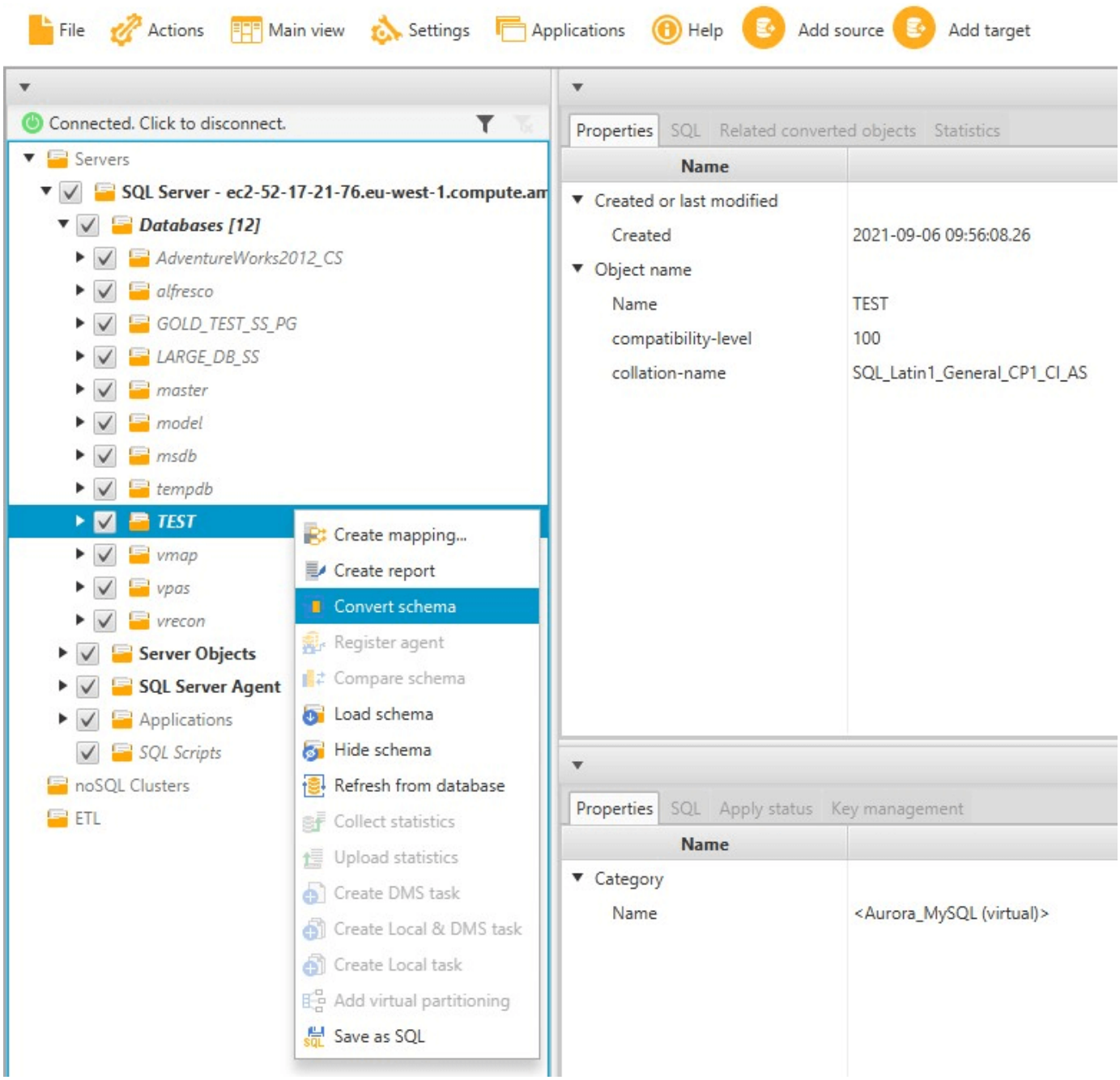
Para convertir su esquema

1. Haga clic en Ver y, a continuación, seleccione Vista principal.



2. En el panel de la izquierda que muestra el esquema de la base de datos de origen, seleccione la casilla de verificación con el nombre del objeto que desea convertir. A continuación, elija este objeto. AWS SCT resalta el nombre del objeto en azul. Abra el menú contextual (clic secundario) para el objeto y seleccione Convertir esquema.

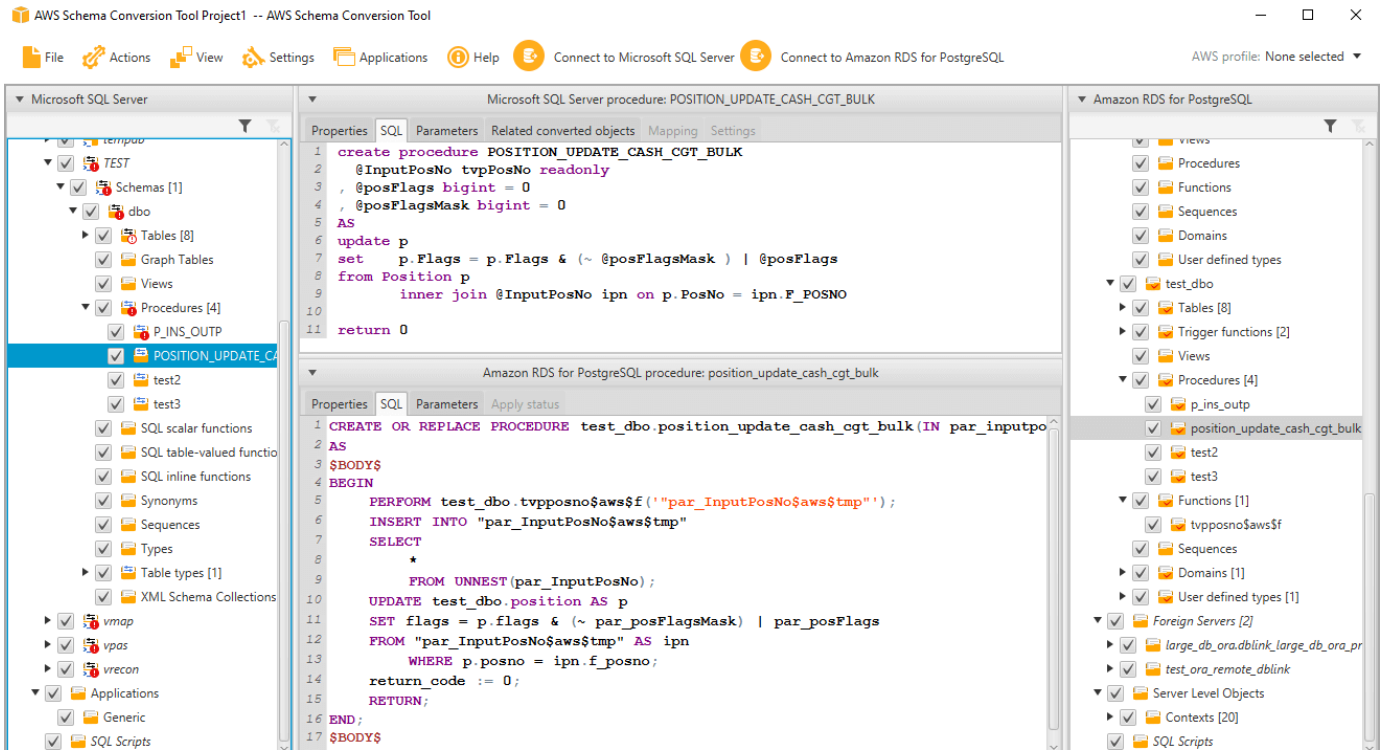
   Para convertir varios objetos de la base de datos, seleccione las casillas de verificación de todos los objetos. A continuación, elija el nodo principal. Por ejemplo, en el caso de las tablas, el nodo principal es Tablas. Asegúrese de que AWS SCT resalta el nombre del nodo principal en azul. Abra el menú contextual (clic secundario) para el nodo principal y seleccione Convertir esquema.



3. Cuando la AWS SCT termina de convertir el esquema, puede ver el esquema propuesto en el panel a la derecha del proyecto.

   En este momento, no se aplica ningún esquema a la instancia de la base de datos de destino. El esquema planificado es parte de su proyecto. Si selecciona un elemento del esquema convertido, podrá ver el comando de esquema previsto en el panel de la parte inferior en el centro de la instancia de base de datos de destino.

Puede editar el esquema en esta ventana. El esquema editado se almacena como parte del proyecto y estará escrito en la instancia de base de datos de destino cuando decida aplicar su esquema convertido.



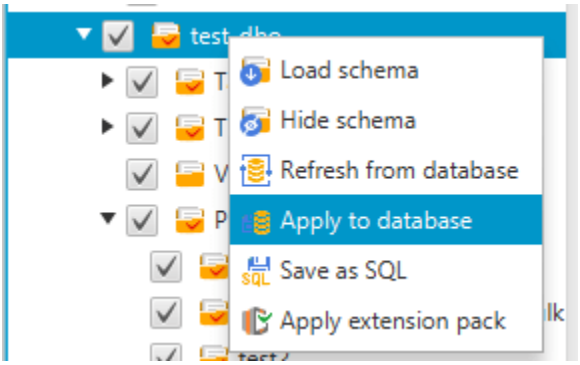
Aplicación del esquema convertido a su instancia de base de datos de destino

Puede aplicar el esquema de base de datos convertido a su instancia de base de datos de de destino. Después de haber aplicado el esquema a su instancia de base de datos de destino, podrá actualizar el esquema en función de los elementos de acción del informe de evaluación de la migración de la base de datos.

> **Warning**
>
> El siguiente procedimiento sobrescribe el esquema de destino existente. Tenga cuidado de no sobrescribir esquemas involuntariamente. Tenga cuidado de no sobrescribir esquemas de su instancia de base de datos de destino que ya haya modificado, o reemplazará dichos cambios.

Para aplicar el esquema de base de datos convertido a su instancia de base de datos de destino

1. Elija Conectar al servidor en la parte superior del panel derecho del proyecto para conectarse a la base de datos de destino. Si está conectado a la base de datos de destino, omita este paso.
2. Seleccione el elemento del esquema del panel derecho del proyecto que indique el esquema previsto para su instancia de base de datos de destino.
3. Abra el menú contextual (clic con el botón secundario) del elemento del esquema y seleccione Apply to database.



   El esquema convertido se aplica a la instancia de base de datos de destino.

Almacenamiento de perfiles de servicio de AWS en AWS SCT

Puede almacenar sus credenciales de AWS en la AWS SCT. AWS SCT utiliza sus credenciales al recurrir a características que se integran con los servicios de AWS. Por ejemplo, AWS SCT se integra con Amazon S3, AWS Lambda, Amazon Relational Database Service (Amazon RDS) y AWS Database Migration Service (AWS DMS).

AWS SCT le solicita sus credenciales de AWS al obtener acceso a una característica que las necesite. Puede almacenar sus credenciales en los ajustes globales de la aplicación. Cuando AWS SCT le pida sus credenciales, puede seleccionar las credenciales almacenadas.

Puede almacenar distintos conjuntos de credenciales de AWS en los ajustes globales de la aplicación. Por ejemplo, puede almacenar un conjunto de credenciales que utilice en situaciones de prueba y un conjunto de credenciales diferente que utilice en situaciones de producción. También puede almacenar diferentes credenciales para diferentes Región de AWS.

Almacenamiento de credenciales de AWS

Utilice el siguiente procedimiento para almacenar las credenciales de AWS de manera global.

Para almacenar credenciales de AWS

1. Inicie AWS Schema Conversion Tool.
2. Abra el menú Configuración y seleccione Configuración global. Aparecerá el cuadro de diálogo Global settings.
3. Elija Perfiles de servicio de AWS y, a continuación, elija Agregar un nuevo perfil de servicio de AWS.
4. Introduzca su información de AWS como sigue.

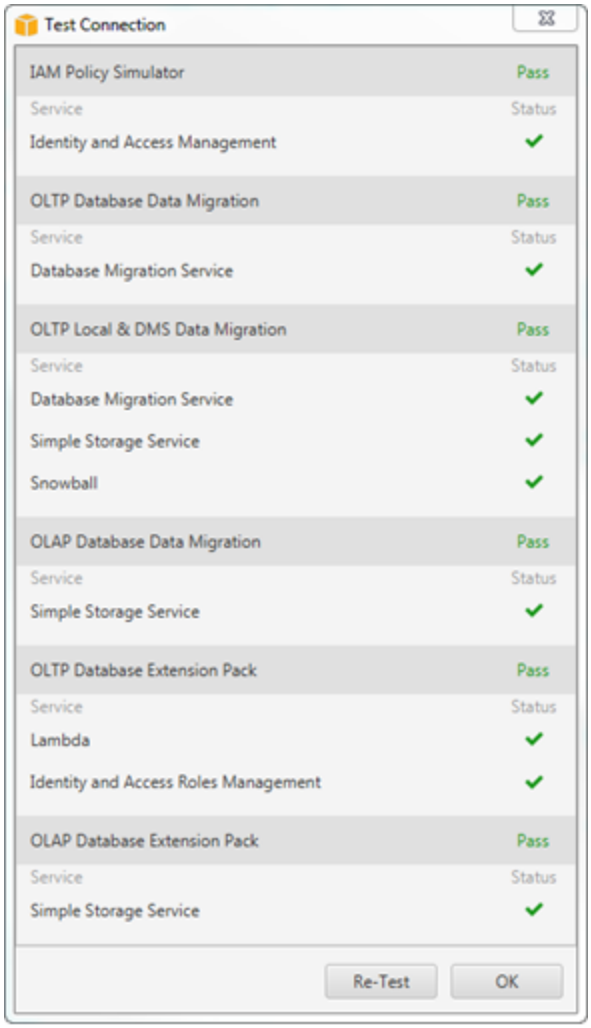
Opción de AWS SCT	Acción
Profile name	Escriba un nombre para su perfil.
Clave de acceso de AWS	Introduzca su clave de acceso de AWS.
Clave secreta de AWS	Introduzca su clave de acceso secreta de AWS. Para obtener más información acerca de las claves de acceso de AWS, consulte Administrar claves de acceso en la Guía del usuario de IAM.
Región	Elija la Región de AWS para su perfil.
Carpeta de bucket de Amazon S3	Elija el bucket de Amazon S3) para su perfil. Solo tiene que especific ar un bucket si está utilizando una característica que se conecte a Amazon S3. Para obtener más información acerca de los privilegios necesarios, consulte Permisos para usar el perfil de servicio de AWS.

Seleccione Use FIPS endpoint for S3 si tiene que cumplir con los requisitos de seguridad del Estándar federal de procesamiento de la información (FIPS, Federal Information Processing Standard). Hay puntos de conexión de FIPS en las siguientes regiones de AWS:

- US East (N. Virginia) Region
- US East (Ohio) Region
- Región del oeste de EE. UU (N. California)
- Región del oeste de EE. UU (Oregon)

5. Seleccione Probar conexión para comprobar que sus credenciales sean correctas y estén activas.

   Aparecerá el cuadro de diálogo Probar conexión. Podrá ver el estado de cada uno de los servicios conectados a su perfil. Pass indica que el perfil puede obtener acceso al servicio correctamente.



6. Tras haber configurado su perfil, seleccione Save para guardar su perfil o Cancel para cancelar los cambios.
7. Seleccione Aceptar para cerrar el cuadro de diálogo Configuración global.

Configurar el perfil predeterminado para un proyecto

Puede configurar el perfil predeterminado para un proyecto de AWS SCT. De este modo, asociará las credenciales de AWS almacenadas en el perfil con el proyecto. Con el proyecto abierto, utilice el siguiente procedimiento para configurar el perfil predeterminado.

Para configurar el perfil predeterminado para un proyecto

1. Inicie AWS Schema Conversion Tool y cree un proyecto nuevo.
2. En el menú Configuración, seleccione Configuración del proyecto. Aparecerá el cuadro de diálogo Configuración del proyecto.
3. Seleccione la pestaña Entorno del proyecto.
4. Seleccione Agregar un perfil de servicio de AWS nuevo para agregar un perfil nuevo. En Perfil de servicio de AWS, seleccione el perfil que quiera asociar con el proyecto.
5. Seleccione Aceptar para cerrar el cuadro de diálogo Configuración del proyecto. También puede seleccionar Cancel para cancelar los cambios.

Permisos para usar el perfil de servicio de AWS

Se necesitan los siguientes permisos necesarios para acceder a su bucket de Amazon S3 desde su perfil de servicio de AWS:

- `s3:PutObject`: para agregar objetos en su bucket de Amazon S3.
- `s3:DeleteObject`: para eliminar la versión nula de un objeto e insertar un marcador de eliminación, que se convierte en la versión actual del objeto.
- `s3:ListBucket`: para devolver hasta 1 000 objetos de su bucket de Amazon S3.
- `s3:GetObject`: para recuperar objetos de su bucket de Amazon S3.

En el siguiente ejemplo de código, se muestra cómo conceder estos permisos a un usuario.

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

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

Utilizar AWS Secrets Manager

AWS SCT puede utilizar las credenciales de base de datos que almacene en AWS Secrets Manager. Puede rellenar todos los valores en el cuadro de diálogo de conexión a la base de datos de Secrets Manager. Para usar Secrets Manager, guarde los perfiles de AWS en AWS Schema Conversion Tool.

Para obtener más información sobre el uso de AWS Secrets Manager, consulte ¿Qué es AWS Secrets Manager? en la Guía del usuario de AWS Secrets Manager. Para obtener más información sobre el almacenamiento de perfiles de AWS, consulte Almacenamiento de perfiles de servicio de AWS en AWS SCT.

Para recuperación las credenciales de la base de datos desde Secrets Manager

1. Inicie AWS Schema Conversion Tool y cree un proyecto nuevo.
2. Elija Agregar origen o Agregar destino para agregar una base de datos nueva al proyecto.
3. Seleccione una plataforma de base de datos y, a continuación, elija Siguiente..
4. En Secreto de AWS, elija el secreto que desea utilizar.
5. Elija Rellenar. A continuación, AWS SCT rellena todos los valores del cuadro de diálogo de conexión a la base de datos.
6. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos.
7. Seleccione Conectar para conectarse a su base de datos.

AWS SCT admite secretos que tienen la siguiente estructura.

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

En esta estructura, los valores `username` y `password` son obligatorios y todos los demás valores son opcionales. Asegúrese de que los valores que almacena en Secrets Manager incluyan todas las credenciales de la base de datos.

Almacenar contraseñas de la base de datos

Puede almacenar un certificado SSL o una contraseña de la base de datos en la caché de la AWS SCT. Para almacenar una contraseña, seleccione Store Password al crear una conexión.

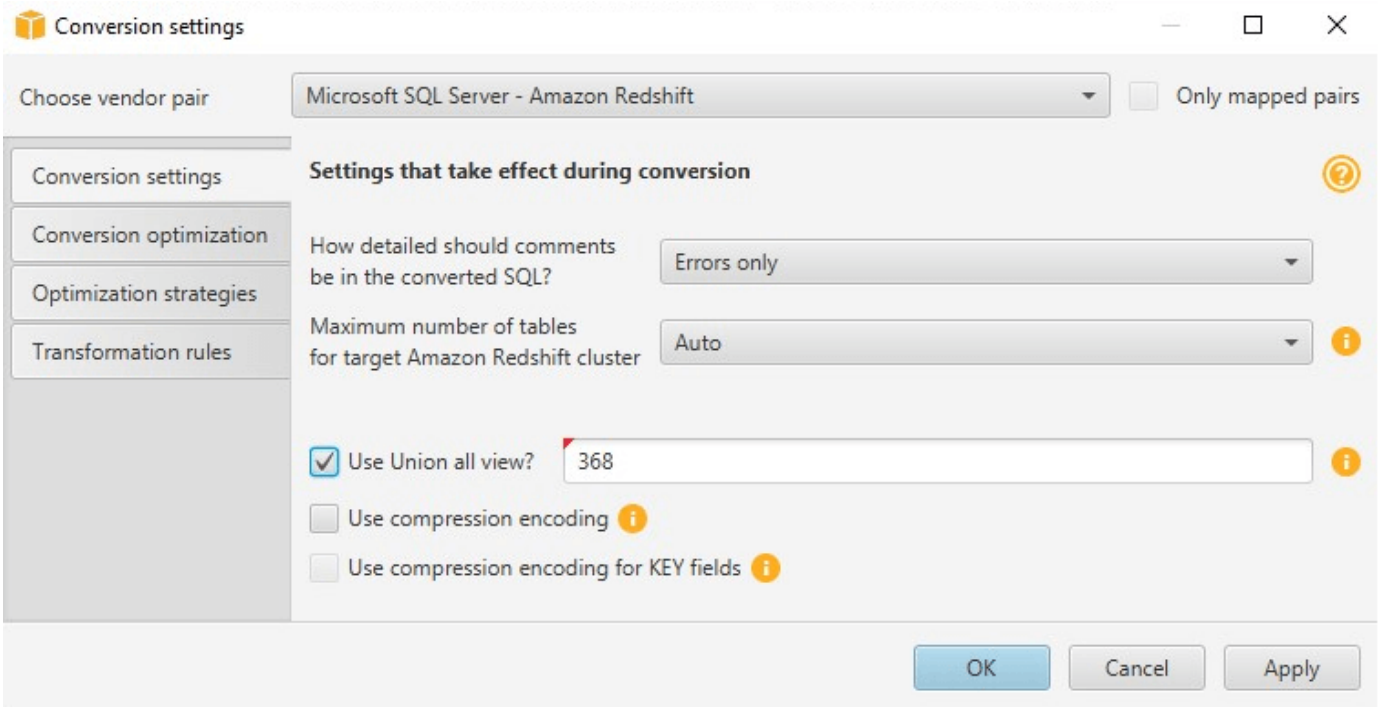
La contraseña se cifra mediante un token generado de forma aleatoria en el archivo `seed.dat`. A continuación, la contraseña se almacena con el nombre del usuario en el archivo en la memoria caché. Si pierde el archivo `seed.dat` o resulta dañado, puede que la contraseña de la base de datos se descifre de forma incorrecta. En este caso, la conexión produce un error.

Utilizar la vista UNION ALL en proyectos con tablas con particiones

Si una tabla de origen está particionada, AWS SCT crea n tablas de destino, donde n es el número de particiones de la tabla de origen. AWS SCT crea una vista UNION ALL en la parte superior de las tablas de destino para representar la tabla de origen. Si utiliza un extractor de datos de AWS SCT para migrar los datos, las particiones de la tabla de origen se extraerán y cargarán en paralelo mediante subtareas independientes.

Para usar la vista UNION ALL en un proyecto

1. Inicie AWS SCT. Cree un proyecto nuevo o abra un proyecto de AWS SCT existente.
2. En el menú Configuración, seleccione Configuración de conversión.
3. Seleccione un par de bases de datos OLAP de la lista de la parte superior.
4. Active la opción ¿Usar vista Union all?



5. Seleccione Aceptar para guardar la configuración y cerrar el cuadro de diálogo Configuración de conversión.

Métodos abreviados de teclado para AWS SCT

A continuación presentamos los métodos abreviados de teclado que puede utilizar con AWS SCT.

Método abreviado de teclado	Descripción
Ctrl+N	Crear un nuevo proyecto.
Ctrl+O	Abrir un proyecto existente.
Ctrl+S	Guardar un proyecto abierto.
Ctrl+W	Crear un nuevo proyecto con el asistente.
Ctrl+M	Crear una evaluación multiservidor nueva.
Ctrl+L	Agregar una base de datos de origen nueva.
Ctrl+R	Agregar una base de datos de destino nueva.

Método abreviado de teclado	Descripción
Ctrl+F3	Cerrar un proyecto abierto.
F1	Abrir la Guía del usuario de AWS SCT.

Introducción a AWS SCT

Puede usar AWS Schema Conversion Tool (AWS SCT) para convertir el esquema de una base de datos de origen. El origen de la base de datos puede hallarse en un motor autoadministrado en ejecución en las instalaciones o en una instancia de Amazon EC2. Puede convertir su esquema de origen en un esquema para cualquier base de datos compatible alojada por AWS. La aplicación AWS SCT ofrece una interfaz de usuario basada en proyectos.

Casi todo el trabajo que realice con AWS SCT comenzará por los siguientes pasos:

1. Instale AWS SCT. Para obtener más información, consulte Instalación, verificación y actualización AWS SCT.
2. Instale un agente de AWS SCT, si es necesario. Los agentes de AWS SCT solo son necesarios para determinadas situaciones de migración como, por ejemplo, entre orígenes y destinos heterogéneos. Para obtener más información, consulte Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift.
3. Familiarícese con la interfaz de usuario de AWS SCT. Para obtener más información, consulte Utilización de la interfaz de usuario de AWS SCT.
4. Cree un proyecto de AWS SCT. Conéctese a las bases de datos de origen y destino. Para obtener más información acerca de la conexión a la base de datos de origen, consulte Orígenes para AWS SCT.
5. Cree reglas de asignación. Para obtener más información acerca de las reglas de asignación, consulte Creación de reglas de asignación en AWS SCT.
6. Ejecutar y consultar el informe de evaluación de la migración de la base de datos. Para obtener más información sobre el informe de evaluación, consulte Creación y revisión del informe de evaluación de la migración de la base de datos.
7. Convertir los esquemas de base de datos de origen. Existen varios aspectos de la conversión que debe tener en cuenta, por ejemplo, qué se debe hacer con los elementos que no se convierten y cómo mapear los elementos que se deben convertir de una forma concreta. Para obtener más información acerca de la conversión de un esquema de origen, consulte Conversión de esquemas de base de datos mediante AWS SCT.

   Si va a convertir un esquema de almacenamiento de datos, también hay aspectos que debe tener en cuenta antes de realizar la conversión. Para obtener más información, consulte Convertir esquemas de almacenamiento de datos a Amazon Redshift mediante AWS SCT.

8. Aplicar la conversión de esquema al destino. Para obtener más información acerca de la aplicación de la conversión de esquemas de origen, consulte Aplicación del código convertido.
9. También puede usar AWS SCT para convertir procedimientos almacenados de SQL y otro código de aplicación. Para obtener más información, consulte Conversión de SQL de las aplicaciones mediante AWS SCT

También puede utilizar AWS SCT para migrar los datos de una base de datos de origen a una base de datos administrada por Amazon. Para ver ejemplos, consulte Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift.

Orígenes para AWS SCT

AWS Schema Conversion Tool (AWS SCT) puede convertir esquemas de las siguientes bases de datos y almacenamientos de datos de origen en una base de datos o almacenamiento de datos de destino. Para obtener información acerca de los permisos, las conexiones y lo que AWS SCT puede convertir para usar con la base de datos o el almacenamiento de datos de destino, consulte los detalles de los temas que se muestran a continuación.

Información de cifrado

Cifrado de las conexiones de Amazon RDS

Bases de datos de origen

- Usar Apache Cassandra como origen
- Usar Azure SQL Database como origen
- Usar IBM Db2 para z/OS como origen
- Usar IBM Db2 LUW como origen
- Usar MySQL como origen
- Usar una base de datos de Oracle como origen
- Uso de PostgreSQL como origen
- Usar SAP ASE (Sybase ASE) como origen
- Uso de SQL Server como origen

Orígenes de almacenamiento de datos

- Uso de Amazon Redshift como origen
- Utilizar Azure Synapse Analytics como origen para
- Usar BigQuery como origen
- Utilizar la base de datos de Greenplum como origen
- Utilizar Netezza como origen
- Utilizar Oracle Data Warehouse como origen
- Utilizar Snowflake como origen
- Utilizar Microsoft SQL Server Data Warehouse como origen

- Utilizar Teradata como origen
- Uso de Vertica como origen

Orígenes de macrodatos

- Usar Apache Hadoop como origen
- Usar Apache Oozie como origen

Cifrado de las conexiones de Amazon RDS y Amazon Aurora en AWS SCT

Para abrir conexiones cifradas a bases de datos de Amazon RDS o Amazon Aurora desde una aplicación, debe importar los certificados raíz de AWS a algún tipo de almacenamiento de claves. Puede descargar los certificados raíz de AWS en Uso de SSL/TLS para cifrar una conexión a una instancia de base de datos de la Guía del usuario de Amazon RDS.

Hay dos opciones disponibles: un certificado raíz que funciona en todas las regiones de AWS y un paquete de certificados que contiene tanto los certificados raíz antiguos como los nuevos.

En función de la opción que desee utilizar, siga los pasos de uno de estos dos procedimientos.

Para importar el certificado o los certificados al almacenamiento del sistema de Windows

1. Descargue uno o varios certificados de una de las siguientes fuentes:

   Para obtener más información sobre la descarga de certificados, consulte Usar SSL/TLS para cifrar una conexión a una instancia de base de datos en la Guía del usuario de Amazon RDS.
2. En la ventana de búsqueda de Windows, introduzca **`Manage computer certificates`**. Cuando se le pregunte si desea permitir que la aplicación realice cambios en su equipo, seleccione Sí.
3. Cuando se abra la ventana de certificados, si es necesario, expanda Certificados: equipo local para ver la lista de certificados. Abra el menú contextual (clic secundario) de Entidades de certificación raíz de confianza y, a continuación, elija Todas las tareas, Importar.
4. Seleccione Siguiente y, a continuación, Examinar y busque el archivo `*.pem` que descargó en el paso 1. Elija Abrir para seleccionar el archivo de certificado, elija Siguiente y, a continuación, elija Terminar.

> **Note**
>
> Para buscar un archivo, cambie el tipo de archivo en la ventana de navegación a Todos los archivos (*.*), ya que `.pem` no es una extensión de certificado estándar.

5. En Microsoft Management Console, expanda Certificados. A continuación, expanda Entidades de certificación raíz de confianza, seleccione Certificados y busque el certificado para confirmar que existe. El nombre del certificado empieza por `Amazon RDS`.
6. Reinicie el equipo.

Para importar el certificado o los certificados al almacén de claves de Java

1. Descargue uno o varios certificados de uno de los siguientes orígenes:

   Para obtener más información sobre la descarga de certificados, consulte Usar SSL/TLS para cifrar una conexión a una instancia de base de datos en la Guía del usuario de Amazon RDS.
2. Si descargó el paquete de certificados, divídalo en archivos de certificados individuales. Para ello, coloque cada bloque de certificados que comience por `-----BEGIN CERTIFICATE-----` y termine por `-----END CERTIFICATE-----` en archivos `*.pem` independientes. Una vez que haya creado un archivo `*.pem` independiente para cada certificado, puede eliminar de forma segura el archivo del paquete de certificados.
3. Abra una ventana de comandos o una sesión de terminal en el directorio en el que descargó el certificado y ejecute el siguiente comando para cada archivo `*.pem` que haya creado en el paso anterior.

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

   Example

   En el ejemplo siguiente se presupone que ha descargado el archivo `eu-west-1-bundle.pem`.

   ```
   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. Agregue el almacén de claves como almacén de confianza en AWS SCT. Para ello, en el menú principal, seleccione Configuración, Configuración global, Seguridad, Almacén de confianza y, a continuación, Seleccionar almacén de confianza existente.

   Tras agregar el almacén de confianza, puede utilizarlo para configurar una conexión con SSL al crear una conexión de AWS SCT a la base de datos. En el cuadro de diálogo Conectar a la base de datos de AWS SCT, elija Usar SSL y elija el almacén de confianza introducido anteriormente.

Uso de Apache Cassandra como origen para AWS SCT

Puede utilizar AWS SCT para convertir espacios de claves de Apache Cassandra a Amazon DynamoDB.

Conexión a Apache Cassandra como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de origen de Apache Cassandra con AWS Schema Conversion Tool.

Para conectar a una base de datos de origen de Apache Cassandra

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Elija Cassandra y, a continuación, seleccione Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:
     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.

- Para introducir manualmente la información de conexión a la base de datos de origen de Cassandra Apache, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del servicio de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Usar SSL	Seleccione esta opción si desea utilizar capa de sockets seguros (SSL) para conectarse a la base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Trust store: el almacén de confianza que se va a utilizar. • Key store: el almacén de claves que se va a utilizar.

Parámetro	Acción
Store password	AWS SCT crea un almacén seguro para almacenar certificados SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Uso de Apache Hadoop como origen para AWS SCT

Puede utilizar la interfaz de la línea de comandos (CLI) de AWS SCT para migrar desde Apache Hadoop a Amazon EMR. AWS SCTutiliza su bucket de Amazon S3 como almacenamiento temporal para sus datos durante la migración.

AWS SCT admite como origen la versión 2.2.0 y posteriores de Apache Hadoop. Además, AWS SCT es compatible con la versión 0.13.0 y posteriores de Apache Hive.

AWS SCT admite como destino la versión 6.3.0 y posteriores de Amazon EMR. Además, AWS SCT admite como destino la versión 2.6.0 y posteriores de Apache Hadoop y la versión 0.13.0 y posteriores de Apache Hive.

Temas

- Requisitos previos para utilizar Apache Hadoop como origen
- Permisos para usar Hive como origen
- Permisos para usar HDFS como origen
- Permisos para usar HDFS como destino
- Conectar a Apache Hadoop como origen
- Conexión a sus servicios Hive y HDFS de origen
- Conexión a Amazon EMR como destino

Requisitos previos para utilizar Apache Hadoop como origen

Estos son los requisitos previos necesarios para conectarse a Apache Hadoop con la CLI de AWS SCT.

- Cree un bucket de Amazon S3 para almacenar datos durante la migración. A continuación, puede copiar los datos a Amazon EMR HDFS o utilizar Amazon S3 como repositorio de datos para sus cargas de trabajo de Hadoop. Para obtener más información, consulte Crear un bucket en la Guía del usuario de Amazon S3.
- Cree un rol AWS Identity and Access Management (de IAM) con la política de `AmazonS3FullAccess`. AWS SCT utiliza este rol de IAM para acceder a su bucket de Amazon S3.
- Anote su clave secreta de AWS y su clave de acceso secreta de AWS. Para obtener más información acerca de las claves de acceso de AWS, consulte Administrar claves de acceso en la Guía del usuario de IAM.
- Cree y configure un clúster de Amazon EMR de destino. Para más información, consulte Introducción a Amazon EMR en la Guía de administración de Amazon EMR.
- Instale la utilidad `distcp` en el clúster Apache Hadoop de origen. Además, instale la utilidad `s3-dist-cp` en el clúster de Amazon EMR de destino. Asegúrese de que los usuarios de la base de datos tengan permisos para ejecutar estas utilidades.
- Configure el archivo `core-site.xml` en el clúster de Hadoop de origen para usar el protocolo s3a. Para ello, establezca el parámetro `fs.s3a.aws.credentials.provider` en uno de los siguientes valores.
  - `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`

  Puede añadir el siguiente código de ejemplo al archivo `core-site.xml`.

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

El ejemplo anterior muestra una de las cuatro opciones de la lista anterior. Si no establece el parámetro `fs.s3a.aws.credentials.provider` en el archivo `core-site.xml`, AWS SCT elige el proveedor automáticamente.

Permisos para usar Hive como origen

Los permisos necesarios para un usuario de Hive como origen son los siguientes:

- Acceso `READ` a las carpetas de datos de origen y al bucket de Amazon S3 de origen
- Acceso `READ+WRITE` a los buckets de Amazon S3 intermedio y de destino

Para aumentar la velocidad de migración, le recomendamos que ejecute la compactación de las tablas de origen transaccionales de ACID.

Los permisos necesarios para un usuario de Amazon EMR como destino son los siguientes:

- Acceso `READ` al bucket de Amazon S3 de destino
- Acceso `READ+WRITE` al bucket de Amazon S3 intermedio
- Acceso `READ+WRITE` a las carpetas HDFS de destino

Permisos para usar HDFS como origen

Los permisos necesarios para un usuario de HDFS como origen son los siguientes:

- `EXECUTE` para el NameNode
- `EXECUTE+READ` para todas las carpetas y archivos de origen que incluya en el proyecto de migración
- `READ+WRITE` para que el directorio `tmp` del NameNode ejecute las tareas de Spark y almacene los archivos antes de la migración a Amazon S3

En HDFS, todas las operaciones requieren un acceso transversal. El acceso transversal exige el permiso `EXECUTE` de todos los componentes existentes de la ruta, excepto el componente final de la ruta. Por ejemplo, para que cualquier operación acceda a `/foo/bar/baz`, el usuario debe tener permiso `EXECUTE` acceder a `/`, `/foo` y `/foo/bar`.

El siguiente ejemplo de código muestra cómo conceder permisos `EXECUTE+READ` para las carpetas y archivos de origen y permisos `READ+WRITE` para el directorio `tmp`.

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

Permisos para usar HDFS como destino

Los permisos necesarios para Amazon EMR HDFS como destino son los siguientes:

- `EXECUTE` para el NameNode del clúster de Amazon EMR de destino
- `READ+WRITE` para las carpetas HDFS de destino en las que almacenará los datos tras la migración

Conectar a Apache Hadoop como origen

Puede usar Apache Hadoop como origen en la versión 1.0.670 o posteriores de AWS SCT. Puede migrar los clústeres de Hadoop a Amazon EMR solo en la interfaz de la línea de comandos (CLI) de AWS SCT. Antes de empezar, familiarícese con la interfaz de la línea de comandos de AWS SCT. Para obtener más información, consulte AWS SCT Referencia CLI.

Para conectarse a Apache Hadoop en la CLI de AWS SCT

1. Cree un script de la CLI de AWS SCT nuevo o edite una plantilla de escenario existente. Por ejemplo, puede descargar y editar la plantilla de `HadoopMigrationTemplate.scts`. Para obtener más información, consulte Obtención de escenarios de la CLI.
2. Configure los ajustes de aplicación de AWS SCT, como la ubicación del controlador y la carpeta de registro.

   Descargue el controlador JDBC necesario y, a continuación, especifique la ubicación en la que se almacena el archivo. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios.

   En el siguiente ejemplo de código, se muestra cómo agregar la ruta de acceso al controlador Apache Hive. Tras ejecutar este ejemplo de código, AWS SCT almacena los archivos de registro en la carpeta `c:\sct`.

   ```
   SetGlobalSettings
   ```

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

Puede usar este ejemplo y los ejemplos siguientes en Windows.

3. Cree un proyecto de AWS SCT nuevo.

   El siguiente ejemplo de código crea el proyecto de `hadoop_emr` en la carpeta `c:\sct`.

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

4. Agregue su clúster de Hadoop de origen al proyecto.

   Utilice el comando `AddSourceCluster` para conectarse al clúster de Hadoop de origen. Proporcione valores para los siguientes parámetros obligatorios: `name`, `host`, `port` y `user`. El resto de los parámetros son opcionales.

   El siguiente ejemplo de código agrega el clúster de Hadoop de origen. En este ejemplo `HADOOP_SOURCE` se establece el nombre del clúster de origen. Utilice este nombre de objeto para añadir los servicios Hive y HDFS al proyecto y crear reglas de asignación.

```
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'
/
```

En el ejemplo anterior, sustituya *hadoop_address* por la dirección IP del clúster de Hadoop. Si es necesario, configure el valor de la opción de puerto. A continuación, sustituya *hadoop_user* y *hadoop_password* por su nombre de usuario de Hadoop y la contraseña de este usuario. En *path\name*, introduzca el nombre y la ruta del archivo PEM del clúster de Hadoop de origen.

5. Guarde el script de la CLI. A continuación, agregue la información de conexión de los servicios Hive y HDFS.

Conexión a sus servicios Hive y HDFS de origen

Puede conectarse a los servicios Hive y HDFS de origen con la CLI de AWS SCT. Para conectarse a Apache Hive, utilice el controlador JDBC de Hive, versión 2.3.4 o posterior. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios.

AWS SCT se conecta a Apache Hive con el usuario del clúster de `hadoop`. Para ello, utilice los comandos `AddSourceClusterHive` y `AddSourceClusterHDFS`. Puede utilizar una de las siguientes opciones.

- Cree un túnel SSH nuevo.

  En `createTunnel`, introduzca **`true`**. En `host`, introduzca la dirección IP interna del servicio Hive o HDFS de origen. En `port`, introduzca el puerto de servicio del servicio Hive o HDFS.

  A continuación, introduzca sus credenciales de Hive o HDFS para `user` y `password`. Para obtener más información sobre los túneles SSH, Configurar un túnel SSH hacia el nodo principal mediante el reenvío dinámico de puertos en la Guía de administración de Amazon EMR.

- Utilice un túnel SSH existente.

  En `host`, introduzca **`localhost`**. En `port`, introduzca el puerto local a partir de los parámetros del túnel SSH.

- Conéctese directamente a sus servicios Hive y HDFS.

  En `host`, introduzca la dirección IP o el nombre de host del servicio Hive o HDFS de origen. En `port`, introduzca el puerto de servicio del servicio Hive o HDFS. A continuación, introduzca sus credenciales de Hive o HDFS para `user` y `password`.

Para conectarse a Hive y HDFS en la CLI de AWS SCT

1. Abra el script de la CLI que incluye la información de conexión del clúster Hadoop de origen. Utilice el nombre del clúster de Hadoop que definió en el paso anterior.
2. Agregue su servicio Hive de origen al proyecto.

   Use el comando `AddSourceClusterHive` para conectar el servicio Hive de origen. Proporcione valores para los siguientes parámetros obligatorios: `user`, `password`, `cluster`, `name` y `port`. El resto de los parámetros son opcionales.

   El siguiente ejemplo de código crea un túnel para que AWS SCT funcione con el servicio Hive. Este servicio Hive de origen se ejecuta en el mismo equipo que AWS SCT. En este ejemplo se utiliza el clúster de origen de `HADOOP_SOURCE` del ejemplo anterior.

```
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'
/
```

   El siguiente ejemplo de código conecta su servicio Hive sin un túnel.

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

   En los ejemplos anteriores, sustituya *`hive_user`* y *`hive_password`* por el nombre del usuario de Hive y la contraseña de este usuario.

A continuación, sustituya *hive_address* y *hive_port* por la dirección IP y el puerto de NameNode del clúster de Hadoop de origen.

En *hive_remote_address*, puede usar el valor predeterminado `127.0.0.1` o la dirección IP de NameNode del servicio Hive de origen.

3. Agregue su servicio HDFS de origen al proyecto.

   Use el comando `AddSourceClusterHDFS` para conectar el servicio HDFS de origen. Proporcione valores para los siguientes parámetros obligatorios: `user`, `password`, `cluster`, `name` y `port`. El resto de los parámetros son opcionales.

   El usuario debe tener los permisos necesarios para migrar datos desde el servicio HDFS de origen. Para obtener más información, consulte Permisos para usar Hive como origen.

   El siguiente ejemplo de código crea un túnel para que AWS SCT funcione con el servicio HDFS. En este ejemplo, se utiliza el clúster de origen de `HADOOP_SOURCE` que creó anteriormente.

```
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'
/
```

   El siguiente ejemplo de código conecta su servicio Apache HDFS sin un túnel.

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

En los ejemplos anteriores, sustituya *hdfs_user* y *hdfs_password* por el nombre del usuario de HDFS y la contraseña de este usuario.

A continuación, sustituya *hdfs_address* y *hdfs_port* por la dirección IP y el puerto de NameNode del clúster de Hadoop de origen.

En *hdfs_remote_address*, puede usar el valor predeterminado `127.0.0.1` o la dirección IP de NameNode del servicio Hive de origen.

4. Guarde el script de la CLI. A continuación, agregue la información de conexión del clúster de Amazon EMR de destino y los comandos de migración.

Conexión a Amazon EMR como destino

Puede conectarse a su clúster de Amazon EMR de destino con la CLI de AWS SCT. Para ello, autorice el tráfico entrante y utilice SSH. En este caso, AWS SCT tiene todos los permisos necesarios para trabajar con su clúster de Amazon EMR. Para obtener más información, consulte Antes de conectar y Conectarse al nodo principal mediante SSH en la Guía de administración de Amazon EMR.

AWS SCT se conecta a Amazon EMR Hive con el usuario del clúster de `hadoop`. Para conectarse a Amazon EMR Hive, utilice el controlador JDBC de Hive, versión 2.6.2.1002 o posterior. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios.

Para conectarse a Amazon EMR en la CLI de AWS SCT

1. Abra el script de la CLI que incluye la información de conexión del clúster Hadoop de origen. Agregue las credenciales de Amazon EMR de destino a este archivo.
2. Agregue su clúster de Amazon EMR de destino al proyecto.

   El siguiente ejemplo de código se conecta al clúster de Amazon EMR de destino. En este ejemplo `HADOOP_TARGET` se establece el nombre del clúster de destino. Utilice este nombre de objeto para añadir los servicios Hive y HDFS y una carpeta del bucket de Amazon S3 al proyecto y crear reglas de asignación.

```
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'
/
```

En el ejemplo anterior, introduzca los nombres de los recursos de AWS y la información de conexión de Amazon EMR. Esto incluye la dirección IP del clúster de Amazon EMR, la clave de acceso de AWS, la clave de acceso secreta de AWS y el bucket de Amazon S3. Si es necesario, configure el valor de la variable de puerto. A continuación, sustituya *emr_user* y *emr_password* por su nombre de usuario de Amazon EMR y la contraseña de este usuario. En *path\name*, introduzca el nombre y la ruta del archivo PEM del clúster de Amazon EMR de origen. Para obtener más información, consulte Descargar archivo PEM para el acceso al clúster de EMR.

3. Agregue su bucket de Amazon S3 al proyecto.

   El siguiente ejemplo de código agrega el bucket de Amazon 3 de destino. En este ejemplo, se utiliza el clúster de HADOOP_TARGET que creó anteriormente.

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

   En el ejemplo anterior, introduzca la clave de acceso de AWS, la clave de acceso secreta de AWS y el bucket de Amazon S3.

4. Agregue su servicio Hive de destino al proyecto.

El siguiente ejemplo de código crea un túnel para que AWS SCT funcione con su servicio Hive de destino. En este ejemplo, se utiliza el clúster de HADOOP_TARGET de destino que creó anteriormente.

```
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'
/
```

En el ejemplo anterior, sustituya *hive_user* y *hive_password* por el nombre del usuario de Hive y la contraseña de este usuario.

A continuación, sustituya *hive_address* por el valor predeterminado 127.0.0.1 o por la dirección IP del NameNode del servicio Hive de destino. A continuación, sustituya *hive_port* por el puerto del servicio Hive de destino.

5. Agregue su servicio HDFS de destino al proyecto.

   El siguiente ejemplo de código crea un túnel para que AWS SCT funcione con el servicio Apache HDFS. En este ejemplo, se utiliza el clúster de HADOOP_TARGET de destino que creó anteriormente.

```
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'
```

```
/
```

En el ejemplo anterior, sustituya *hdfs_user* y *hdfs_password* por el nombre del usuario de HDFS y la contraseña de este usuario.

A continuación, sustituya *hdfs_address* y *hdfs_port* por la dirección IP privada y el puerto del NameNode del servicio HDFS de destino.

6. Guarde el script de la CLI. A continuación, agregue reglas de asignación y comandos de migración. Para obtener más información, consulte Migración de Apache Hadoop a Amazon EMR con .

Usar Apache Oozie como origen para AWS SCT

Puede utilizar la interfaz de la línea de comandos (CLI) de AWS SCT para convertir los flujos de trabajo de Apache Oozie en. AWS Step Functions Tras migrar las cargas de trabajo de Apache Hadoop a Amazon EMR, puede utilizar un servicio nativo en la Nube de AWS para organizar sus trabajos. Para obtener más información, consulte Usar Apache Hadoop como origen.

AWS SCT convierte sus flujos de trabajo de Oozie a AWS Step Functions y utiliza AWS Lambda para simular características que AWS Step Functions no admite. Además, AWS SCT convierte las propiedades del trabajo de Oozie a AWS Systems Manager.

Para convertir los flujos de trabajo de Apache Oozie, utilice la versión 1.0.671 o posterior de AWS SCT. Además, familiarícese con la interfaz de la línea de comandos de AWS SCT. Para obtener más información, consulte AWS SCT Referencia CLI.

Requisitos previos para utilizar Apache Oozie como origen

Estos son los requisitos previos necesarios para conectarse a Apache Oozie con la CLI de AWS SCT.

- Cree un bucket de Amazon S3 para almacenar las definiciones de las máquinas de estado. Puede usar estas definiciones para configurar sus máquinas de estado. Para obtener más información, consulte Crear un bucket en la Guía del usuario de Amazon S3.
- Cree un rol AWS Identity and Access Management (de IAM) con la política de `AmazonS3FullAccess`. AWS SCT utiliza este rol de IAM para acceder a su bucket de Amazon S3.

- Anote su clave secreta de AWS y su clave de acceso secreta de AWS. Para obtener más información acerca de las claves de acceso de AWS, consulte Administrar claves de acceso en la Guía del usuario de IAM.
- Guarde las credenciales de AWS y la información sobre el bucket de Amazon S3 en el perfil de servicios de AWS en la configuración global de la aplicación. A continuación, AWS SCT utiliza este perfil de servicios de AWS para trabajar con sus recursos de AWS. Para obtener más información, consulte Almacenamiento de perfiles de servicio de AWS en AWS SCT.

Para trabajar con los flujos de trabajo de Apache Oozie de origen, AWS SCT requiere la estructura específica de los archivos de origen. Cada una de las carpetas de la aplicación debe incluir el archivo `job.properties`. Este archivo incluye pares clave-valor de las propiedades del trabajo. Además, cada una de las carpetas de la aplicación debe incluir el archivo `workflow.xml`. Este archivo describe los nodos de acción y los nodos de flujo de control del flujo de trabajo.

Conectar a Apache Oozie como origen

Utilice el siguiente procedimiento para conectarse a los archivos de origen de Apache Oozie.

Para conectarse a Apache Oozie en la CLI de AWS SCT

1. Cree un script de la CLI de AWS SCT nuevo o edite una plantilla de escenario existente. Por ejemplo, puede descargar y editar la plantilla de `OozieConversionTemplate.scts`. Para obtener más información, consulte Obtención de escenarios de la CLI.
2. Configure la configuración de aplicación de AWS SCT.

   El siguiente ejemplo de código guarda la configuración de aplicación y permite almacenar las contraseñas en el proyecto. Puede utilizar esta configuración guardada en otros proyectos.

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

3. Cree un proyecto de AWS SCT nuevo.

   El siguiente ejemplo de código crea el proyecto de `oozie` en la carpeta `c:\sct`.

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

4. Agregue la carpeta con los archivos de Apache Oozie de origen al proyecto mediante el comando `AddSource`. Utilice el valor `APACHE_OOZIE` para el parámetro `vendor`. Además, proporcione valores para los siguientes parámetros obligatorios: `name` y `mappingsFolder`.

   El siguiente ejemplo de código agrega Apache Oozie como origen al proyecto de AWS SCT. En este ejemplo, se crea un objeto de origen con el nombre`OOZIE`. Utilice este nombre de objeto para añadir reglas de asignación. Tras ejecutar este ejemplo de código, AWS SCT utiliza la carpeta `c:\oozie`para cargar los archivos de origen en el proyecto.

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

   Puede usar este ejemplo y los ejemplos siguientes en Windows.

5. Conéctese a sus archivos de Apache Oozie de origen mediante el comando `ConnectSource`. Utilice el nombre del objeto de origen que definió en el paso anterior.

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

6. Guarde el script de la CLI. A continuación, agregue la información de conexión para su servicio AWS Step Functions.

Permisos para usar funciones de AWS Lambda en el paquete de extensión

Para las funciones de origen fuente que AWS Step Functions no admite, AWS SCT crea un paquete de extensión. Este paquete de extensión incluye funciones de AWS Lambda que simulan las funciones de origen.

Para usar este paquete de extensión, cree un rol de AWS Identity and Access Management (IAM) con los siguientes permisos.

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

Para aplicar el paquete de extensión, AWS SCT necesita un rol de IAM con los siguientes permisos.

```
{
    "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"
            ]
        }
    ]
}
```

Conexión a AWS Step Functions como origen

Utilice el siguiente procedimiento para conectarse a AWS Step Functions como origen.

Para conectarse a AWS Step Functions en la CLI de AWS SCT

1. Abra el script de la CLI que incluye la información de conexión de los archivos de origen de Apache Oozie.
2. Agregue la información sobre su origen de migración al proyecto de AWS SCT el comando `AddTarget`. Utilice el valor `STEP_FUNCTIONS` para el parámetro `vendor`. Además, proporcione valores para los siguientes parámetros obligatorios: `name` y `profile`.

   El siguiente ejemplo de código agrega AWS Step Functions como origen a su proyecto de AWS SCT. En este ejemplo, se crea un objeto de origen con el nombre`AWS_STEP_FUNCTIONS`. Utilice este nombre de objeto al crear reglas de asignación. Además, en este ejemplo se utiliza el perfil de servicios de AWS SCT que creó en el paso de requisitos previos. Sustituya *`profile_name`* por el nombre del perfil.

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

   Si no usa el perfil de servicios de AWS, proporcione valores para los siguientes parámetros obligatorios:`accessKey`, `secretKey`, `awsRegion` y `s3Path`. Utilice estos parámetros para especificar la clave de acceso secreta de AWS, la clave secreta de AWS, la Región de AWS y la ruta a su bucket de Amazon S3.
3. Conéctese a AWS Step Functions mediante el comando `ConnectTarget`. Utilice el nombre del objeto de origen que definió en el paso anterior.

   El siguiente ejemplo de código se conecta al objeto de destino de `AWS_STEP_FUNCTIONS` utilizando su perfil de servicios de AWS. Sustituya *`profile_name`* por el nombre del perfil.

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

4. Guarde el script de la CLI. A continuación, añada reglas de asignación y comandos de migración. Para obtener más información, consulte Conversión de Apache Oozie a AWS Step Functions .

Usar Azure SQL Database como origen para AWS SCT

Puede usar AWS SCT para convertir esquemas, objetos de código y código de aplicación de Azure SQL Database a los siguientes destinos:

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

Temas

- Privilegios para Azure SQL Database como origen
- Conectar a Azure SQL Database como origen

Privilegios para Azure SQL Database como origen

A continuación se enumeran los privilegios necesarios para Azure SQL Database como origen:

- VIEW DEFINITION
- VIEW DATABASE STATE

Repita la concesión para cada base de datos cuyo esquema vaya a convertir.

Los privilegios necesarios para las bases de datos de MySQL y PostgreSQL de destino se describen en las siguientes secciones.

- Privilegios para MySQL como base de datos de destino
- Privilegios para PostgreSQL como base de datos de destino

Conectar a Azure SQL Database como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de Azure SQL Database de origen con AWS Schema Conversion Tool.

Para conectar a Azure SQL Database como base de datos de origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Seleccione Azure SQL Database y, a continuación, seleccione Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión a la base de datos de origen de Azure SQL Database, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del servicio de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.

Parámetro	Acción
Database	Introduzca el nombre de la base de datos a la que se va a conectar.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Store password	AWS SCT crea un almacén seguro para almacenar certificados SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Usar IBM Db2 para z/OS como origen para AWS SCT

Puede usar AWS SCT para convertir esquemas, objetos de código y código de aplicación de IBM Db2 para z/OS a los siguientes destinos.

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

Requisitos previos para Db2 para z/OS como base de datos de origen

La versión de base de datos de nivel de función 100 de la versión 12 de IBM Db2 para z/OS no admite la mayoría de las capacidades nuevas de la versión 12 de IBM Db2 para z/OS. Esta versión de base de datos proporciona soporte para la solución alternativa para la versión 11 de Db2 y el intercambio de datos con la versión 11 de Db2. Para evitar la conversión de características no compatibles de la versión 11 de Db2, le recomendamos que utilice como origen un nivel de función 500 o superior de base de datos de IBM Db2 para z/OS como origen para AWS SCT.

Puede utilizar el siguiente ejemplo de código para comprobar la versión de la base de datos de IBM Db2 para z/OS de origen.

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

Asegúrese de que este código devuelva la versión `DSN12015` o posterior.

Puede utilizar el siguiente ejemplo de código para comprobar el valor del registro especial `APPLICATION COMPATIBILITY` en su base de datos de IBM Db2 para z/OS de origen.

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

Asegúrese de que este código devuelva la versión `V12R1M500` o posterior.

Privilegios para Db2 para z/OS como base de datos de origen

Los privilegios necesarios para conectarse a una base de datos de Db2 para z/OS y leer los catálogos y tablas del sistema son los siguientes:

- SELECT ON SYSIBM.LOCATIONS
- SELECT ON SYSIBM.SYSCHECKS
- SELECT ON SYSIBM.SYSCOLUMNS
- SELECT ON SYSIBM.SYSDATABASE
- SELECT ON SYSIBM.SYSDATATYPES
- SELECT ON SYSIBM.SYSDUMMY1
- SELECT ON SYSIBM.SYSFOREIGNKEYS
- SELECT ON SYSIBM.SYSINDEXES

- SELECT ON SYSIBM.SYSKEYCOLUSE
- SELECT ON SYSIBM.SYSKEYS
- SELECT ON SYSIBM.SYSKEYTARGETS
- SELECCIONE EN SYSIBM.SYSJAROBJECTS
- SELECT ON SYSIBM.SYSPACKAGE
- SELECT ON SYSIBM.SYSPARMS
- SELECT ON SYSIBM.SYSRELS
- SELECT ON SYSIBM.SYSROUTINES
- SELECT ON SYSIBM.SYSSEQUENCES
- SELECT ON SYSIBM.SYSSEQUENCESDEP
- SELECT ON SYSIBM.SYSSYNONYMS
- SELECT ON SYSIBM.SYSTABCONST
- SELECT ON SYSIBM.SYSTABLES
- SELECT ON SYSIBM.SYSTABLESPACE
- SELECT ON SYSIBM.SYSTRIGGERS
- SELECT ON SYSIBM.SYSVARIABLES
- SELECT ON SYSIBM.SYSVIEWS

Para convertir tablas de Db2 para z/OS en tablas particionadas de PostgreSQL, recopile estadísticas sobre los espacios de tabla y las tablas del base de datos mediante la utilidad RUNSTATS que se muestra a continuación.

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

En el ejemplo anterior, sustituya el marcador de posición *YOURDB* por el nombre de la base de datos de origen.

Conectar a Db2 para z/OS como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de Db2 para z/OS de origen con AWS SCT.

Para conectarse a una base de datos de IBM Db2 para z/OS de origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Elija Db2 para z/OS y, a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión a la base de datos de IBM Db2 para z/OS de origen, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.

Parámetro	Acción
Location	Introduzca el nombre único de la ubicación de Db2 a la que desea obtener acceso.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseñ a de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Usar SSL	Seleccione esta opción si desea utilizar capa de sockets seguros (SSL) para conectarse a la base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Trust store: la ubicación de un almacén de confianza que contenga certificados. Para que esta ubicación aparezca en aquí, añádala en Configuración global.
Store password	AWS SCT crea un almacén seguro para almacenar certifica dos SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.

Parámetro	Acción
Db2 for z/OS driver path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Privilegios para MySQL como base de datos de destino

Los privilegios necesarios para MySQL como destino se enumeran a continuación:

- CREATE ON *.*
- ALTER ON *.*
- DROP ON *.*
- INDEX ON *.*
- REFERENCES ON *.*
- SELECT ON *.*
- CREATE VIEW ON *.*
- SHOW VIEW ON *.*
- TRIGGER ON *.*
- CREATE ROUTINE ON *.*
- ALTER ROUTINE ON *.*
- EXECUTE ON *.*
- SELECT ON mysql.proc

- INSERT, UPDATE ON AWS_DB2ZOS_EXT.*
- INSERT, UPDATE, DELETE ON AWS_DB2ZOS_EXT_DATA*.
- CREATE TEMPORARY TABLES ON AWS_DB2ZOS_EXT_DATA.*

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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';
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. Por último, sustituya *your_password* por una contraseña segura.

Para usar Amazon RDS para MySQL o Aurora MySQL como destino, establezca el parámetro `log_bin_trust_function_creators` en true y el parámetro `character_set_server` en `latin1`. Para configurar estos parámetros, cree un grupo de parámetros de base de datos nuevo o modifique uno existente.

Para usar Aurora MySQL como destino, establezca el parámetro `log_bin_trust_function_creators` en true y el parámetro `character_set_server` en `latin1`. Establezca el parámetro `lower_case_table_names` en true. Para configurar estos parámetros, cree un grupo de parámetros de base de datos nuevo o modifique uno existente.

Privilegios para PostgreSQL como base de datos de destino

Para utilizar PostgreSQL como destino, AWS SCT requiere el privilegio `CREATE ON DATABASE`. Asegúrese de conceder este privilegio a cada base de datos PostgreSQL de destino.

Para utilizar Amazon RDS para PostgreSQL como destino, AWS SCT requiere el privilegio `rds_superuser`.

Para usar los sinónimos públicos convertidos, cambie la ruta de búsqueda predeterminada de la base de datos a `"$user", public_synonyms, public`.

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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;
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. A continuación, sustituya *db_name* por el nombre de la base de datos de destino. Por último, sustituya *your_password* por una contraseña segura.

En PostgreSQL, solo el propietario de un esquema o un `superuser` puede anular un esquema. El propietario puede eliminar un esquema y todos los objetos que incluye este esquema, aunque el propietario del esquema no sea propietario de algunos de los objetos.

Si utiliza distintos usuarios para convertir y aplicar diferentes esquemas a la base de datos de destino, es posible que aparezca un mensaje de error cuando AWS SCT no pueda anular un sistema. Para evitar este mensaje de error, utilice el rol de `superuser`.

Configuración de conversión de Db2 para z/OS a PostgreSQL

Para editar la configuración de conversión de Db2 para z/OS a PostgreSQL, seleccione Configuración y, a continuación, elija Configuración de conversión. En la lista superior, elija Db2 para z/OS y, a continuación, Db2 para z/OS – PostgreSQL o Db2 para z/OS – Amazon Aurora (compatible con PostgreSQL). AWS SCT muestra todos los ajustes disponibles para la conversión de IBM Db2 para z/OS a PostgreSQL.

La configuración de conversión de Db2 para z/OS a PostgreSQL en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.

- Generar nombres únicos para las restricciones de la base de datos de destino.

  En PostgreSQL, todos los nombres de restricciones que utilice deben ser únicos. AWS SCT puede generar nombres únicos para las restricciones en el código convertido añadiendo un prefijo con el nombre de la tabla al nombre de la restricción. Para asegurarse de que AWS SCT genera nombres únicos para sus restricciones, seleccione Generar nombres únicos para las restricciones.

- Conservar el formato de los nombres de las columnas, las expresiones y las cláusulas de las instrucciones DML en el código convertido.

  AWS SCT puede conservar el diseño de los nombres de las columnas, las expresiones y las cláusulas de las instrucciones DML en la misma posición y orden que en el código fuente. Para ello, seleccione Sí en Conservar el formato de los nombres de las columnas, las expresiones y las cláusulas de las instrucciones DML.

- Para excluir las particiones de tabla del ámbito de conversión.

  AWS SCT puede omitir todas las particiones de una tabla de origen durante la conversión. Para ello, seleccione Excluir las particiones de tabla del ámbito de conversión.

- Usar el particionado automático para las tablas particionadas por tamaño.

  Para la migración de datos, AWS SCT puede particionar automáticamente todas las tablas que tengan un tamaño superior al especificado. Para usar esta opción, seleccione Forzar la partición de tablas con un tamaño superior a e introduzca el tamaño de las tablas en gigabytes. A continuación, introduzca el número de particiones. AWS SCT tiene en cuenta el tamaño del dispositivo de almacenamiento de acceso directo (DASD) de la base de datos de origen cuando activa esta opción.

AWS SCT puede determinar el número de particiones de forma automática. Para ello, seleccione Aumentar el número de particiones proporcionalmente e introduzca el número máximo de particiones.

- Devolver conjuntos de resultados dinámicos como una matriz de valores del tipo de datos refcursor.

  AWS SCT puede convertir los procedimientos de origen que devuelven conjuntos de resultados dinámicos en procedimientos que tienen una matriz de refcursors abiertos como parámetro de salida adicional. Para ello, seleccione Utilizar una matriz de refcursos para devolver todos los conjuntos de resultados dinámicos.
- Especificar el estándar que se utilizará para la conversión de valores de fecha y hora en representaciones de cadenas.

  AWS SCT puede convertir valores de fecha y hora en representaciones de cadena mediante uno de los formatos industriales compatibles. Para ello, seleccione Usar representaciones en cadena de valores de fecha o Usar representaciones en cadena de valores de hora. A continuación, elija uno de los siguientes estándares.
  - Organización Internacional de Normalización (ISO)
  - Estándar europeo de IBM (EUR)
  - Estándar estadounidense de IBM (EE. UU.)
  - Estándar industrial japonés Christian Era (JIS)

Usar IBM Db2 LUW como origen para AWS SCT

Puede usar AWS SCT para convertir esquemas objetos de código en el lenguaje SQL y código de aplicación de IBM Db2 para Linux, Unix y Windows (Db2 LUW) a los siguientes destinos.

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

AWS SCT admite como origen las versiones 9.1, 9.5, 9.7, 10.1, 10.5, 11.1 Y 11.5 de Db2 LUW.

Privilegios para DbB2 LUW como origen

Los privilegios necesarios para conectarse a una base de datos de DB2LUW, comprobar los privilegios disponibles y leer los metadatos de esquema de un origen son los siguientes:

- Privilegio necesario para establecer una conexión:
  - CONNECT ON DATABASE
- Privilegio necesario para ejecutar instrucciones SQL:
  - EXECUTE ON PACKAGE NULLID.SYSSH200
- Privilegios necesarios para obtener información de nivel de instancia:
  - EXECUTE ON FUNCTION SYSPROC.ENV_GET_INST_INFO
  - SELECT ON SYSIBMADM.ENV_INST_INFO
  - SELECT ON SYSIBMADM.ENV_SYS_INFO
- Privilegios necesarios para comprobar los privilegios concedidos a través de roles, grupos y autoridades:
  - EXECUTE ON FUNCTION SYSPROC.AUTH_LIST_AUTHORITIES_FOR_AUTHID
  - EXECUTE ON FUNCTION SYSPROC.AUTH_LIST_GROUPS_FOR_AUTHID
  - EXECUTE ON FUNCTION SYSPROC.AUTH_LIST_ROLES_FOR_AUTHID
  - SELECT ON SYSIBMADM.PRIVILEGES
- Privilegios necesarios en los catálogos y las tablas del sistema:
  - SELECT ON SYSCAT.ATTRIBUTES
  - SELECT ON SYSCAT.CHECKS
  - SELECT ON SYSCAT.COLIDENTATTRIBUTES
  - SELECT ON SYSCAT.COLUMNS
  - SELECT ON SYSCAT.DATAPARTITIONEXPRESSION
  - SELECT ON SYSCAT.DATAPARTITIONS
  - SELECT ON SYSCAT.DATATYPEDEP
  - SELECT ON SYSCAT.DATATYPES
  - SELECT ON SYSCAT.HIERARCHIES
  - SELECT ON SYSCAT.INDEXCOLUSE
  - SELECT ON SYSCAT.INDEXES
  - SELECT ON SYSCAT.INDEXPARTITIONS

- SELECT ON SYSCAT.KEYCOLUSE
- SELECT ON SYSCAT.MODULEOBJECTS
- SELECT ON SYSCAT.MODULES
- SELECT ON SYSCAT.NICKNAMES
- SELECT ON SYSCAT.PERIODS
- SELECT ON SYSCAT.REFERENCES
- SELECT ON SYSCAT.ROUTINEPARMS
- SELECT ON SYSCAT.ROUTINES
- SELECT ON SYSCAT.ROWFIELDS
- SELECT ON SYSCAT.SCHEMATA
- SELECT ON SYSCAT.SEQUENCES
- SELECCIONAR EN SYSCAT.TABCONST
- SELECCIONAR EN SYSCAT.TABLES
- SELECCIONE EN SYSCAT.TRIGGERS
- SELECCIONE EN SYSCAT.VARIABLEDEP
- SELECT ON SYSCAT.VARIABLES
- SELECT ON SYSCAT.VIEWS
- SELECT ON SYSIBM.SYSDUMMY1

- Para ejecutar instrucciones SQL, la cuenta de usuario necesita un privilegio para utilizar al menos una de las cargas de trabajo habilitadas en la base de datos. Si ninguna de las cargas de trabajo está asignada al usuario, asegúrese de este puede obtener acceso a la carga de trabajo de usuario predeterminada:
  - USAGE ON WORKLOAD SYSDEFAULTUSERWORKLOAD

Para ejecutar consultas, debe crear un espacio de tabla temporal del sistema con un tamaño de página de 8 K, 16 K y 32 K, si no existen. Para crear los espacios de tabla temporales, ejecute los siguientes scripts.

```
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;
```

Conectar a Db2 LUW como origen

Utilice el siguiente procedimiento para conectarse a su base de datos Db2 LUW de origen con la AWS Schema Conversion Tool.

Para conectarse a una base de datos de origen de Db2 LUW

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Elija Db2 LUW y, a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.

4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.

   - Para introducir manualmente la información de conexión a la base de datos de origen de IBM Db2 LUW, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
Database	Escriba el nombre de la base de datos de Db2 LUW.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.

Parámetro	Acción
Usar SSL	Seleccione esta opción si desea utilizar capa de sockets seguros (SSL) para conectarse a la base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Trust store: la ubicación de un almacén de confianza que contenga certificados. Para que esta ubicación aparezca en aquí, añádala en Configuración global.
Store password	AWS SCT crea un almacén seguro para almacenar certifica dos SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.
Db2 LUW driver path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Convertir Db2 LUW a Amazon RDS para PostgreSQL o Amazon Aurora PostgreSQL - Compatible Edition

Cuando migra IBM Db2 LUW a PostgreSQL, AWS SCT puede convertir varias instrucciones desencadenadoras utilizadas con Db2 LUW. Entre estas instrucciones de activación se incluyen las siguientes:

- Eventos de activación: los eventos de activación INSERT, DELETE y UPDATE especifican que la acción activada debe ejecutarse siempre que el evento se aplica a la tabla o vista de asuntos. Puede especificar cualquier combinación de los eventos INSERT, DELETE y UPDATE, pero puede especificar cada evento solo una vez. AWS SCT admite eventos desencadenadores individuales y múltiples. En el caso de los eventos, PostgreSQL tiene prácticamente la misma funcionalidad.
- Evento OF COLUMN: se puede especificar un nombre de columna desde una tabla base. El disparador se activa únicamente cuando se actualiza una columna que aparece en la lista de nombres de columnas. PostgreSQL tiene la misma funcionalidad.
- Disparadores de instrucciones: estos disparadores indican que la acción se aplica una sola vez en toda la instrucción. Este tipo de granularidad no se puede especificar con los disparadores BEFORE o INSTEAD OF. Si se especifica, se activará un disparador UPDATE o DELETE aunque no haya ninguna fila afectada. PostgreSQL también cuenta con esta funcionalidad y la declaración de los disparadores de instrucciones es idéntica en PostgreSQL y Db2 LUW.
- Cláusulas de referencia: estas cláusulas especifican los nombres de correlación de las variables de transición y los nombres de las tablas de transición. Los nombres de correlación identifican una fila concreta del conjunto de filas afectadas por la operación SQL de activación. Los nombres de las tablas identifican el conjunto completo de filas afectadas. Cada una de las filas afectadas por una operación SQL de activación está disponible para la acción activada al asignar a las columnas los nombres de correlación especificados. PostgreSQL no admite esta funcionalidad y solo utiliza el nombre de correlación NEW u OLD.
- ACTIVADORES INSTEAD OF: AWS SCT los admite.

Convertir tablas particionadas de DB2 LUW a tablas particionadas de PostgreSQL versión 10

AWS SCT puede convertir tablas de Db2 LUW en tablas particionadas de PostgreSQL 10. Existen varias restricciones al convertir una tabla particionada de Db2 LUW en PostgreSQL:

- Puede crear una tabla particionada con una columna que admita valores null en Db2 LUW y puede especificar una partición para almacenar los valores NULL. Sin embargo, PostgreSQL no admite valores NULL con particiones RANGE.
- Db2 LUW puede utilizar una cláusula INCLUSIVE o EXCLUSIVE para establecer los valores límite del intervalo. PostgreSQL solo admite INCLUSIVE para el límite inicial y EXCLUSIVE para el límite final. El nombre de la partición convertida tiene el formato <nombre_tabla_original>_<nombre_partición_original>.

- Puede crear claves principales o únicas para tablas particionadas de Db2 LUW. PostgreSQL requiere que se creen claves primarias o únicas para cada partición directamente. Las restricciones de claves principales o únicas deben eliminarse de la tabla principal. El nombre de la clave convertida tiene el formato <nombre_clave_original>_<nombre_partición_original>.
- Puede crear una restricción de clave externa que tenga como origen o destino una tabla particionada de Db2 LUW. Sin embargo, PostgreSQL no admite referencias de clave externa en tablas particionadas. PostgreSQL tampoco admite las referencias de clave externa entre una tabla particionada y otra tabla.
- Puede crear un índice en una tabla particionada en Db2 LUW. Sin embargo, PostgreSQL requiere que se cree un índice para cada partición directamente. Los índices deben eliminarse de la tabla principal. El nombre del índice convertido tiene el formato <nombre_índice_original>_<nombre_partición_original>.
- Debe definir los disparadores de fila en las particiones individuales, no en la tabla particionada. Los disparadores deben eliminarse de la tabla principal. El nombre del disparador convertido tiene el formato <nombre_disparador_original>_<nombre_partición_original>.

Privilegios para PostgreSQL como destino

Para utilizar PostgreSQL como destino, AWS SCT requiere el privilegio `CREATE ON DATABASE`. Asegúrese de conceder este privilegio a cada base de datos PostgreSQL de destino.

Para usar los sinónimos públicos convertidos, cambie la ruta de búsqueda predeterminada de la base de datos a `"$user", public_synonyms, public`.

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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;
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. A continuación, sustituya *db_name* por el nombre de la base de datos de destino. Por último, sustituya *your_password* por una contraseña segura.

En PostgreSQL, solo el propietario de un esquema o un `superuser` puede anular un esquema. El propietario puede eliminar un esquema y todos los objetos que incluye este esquema, aunque el propietario del esquema no sea propietario de algunos de los objetos.

Si utiliza distintos usuarios para convertir y aplicar diferentes esquemas a la base de datos de destino, es posible que aparezca un mensaje de error cuando AWS SCT no pueda anular un sistema. Para evitar este mensaje de error, utilice el rol de `superuser`.

Convertir Db2 LUW a Amazon RDS para MySQL o Amazon Aurora MySQL.

Al convertir una base de datos de IBM Db2 LUW a RDS para MySQL o Amazon Aurora MySQL, tenga en cuenta lo siguiente.

Privilegios para MySQL como destino

Los privilegios necesarios para MySQL como destino se enumeran a continuación:

- CREATE ON *.*
- ALTER ON *.*
- DROP ON *.*
- INDEX ON *.*
- REFERENCES ON *.*
- SELECT ON *.*
- CREATE VIEW ON *.*
- SHOW VIEW ON *.*
- TRIGGER ON *.*
- CREATE ROUTINE ON *.*
- ALTER ROUTINE ON *.*
- EXECUTE ON *.*
- SELECT ON mysql.proc
- INSERT, UPDATE ON AWS_DB2_EXT.*
- INSERT, UPDATE, DELETE ON AWS_DB2_EXT_DATA*.
- CREATE TEMPORARY TABLES ON AWS_DB2_EXT_DATA.*

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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';
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. Por último, sustituya *your_password* por una contraseña segura.

Para usar Amazon RDS para MySQL o Aurora MySQL como destino, establezca el parámetro `lower_case_table_names` en 1. Este valor significa que el servidor MySQL gestiona los identificadores de nombres de objetos como tablas, índices, disparadores y bases de datos sin distinguir entre mayúsculas y minúsculas. Si ha activado el registro binario en la instancia de destino, establezca el parámetro `log_bin_trust_function_creators` en 1. En este caso, no es necesario utilizar las características `DETERMINISTIC`, `READS SQL DATA` o `NO SQL` para crear funciones almacenadas. Para configurar estos parámetros, cree un grupo de parámetros de base de datos nuevo o modifique uno existente.

Utilizar MySQL como origen para AWS SCT

Puede usar AWS SCT para convertir esquemas, objetos de código de bases de datos y código de aplicación de MySQL a los siguientes destinos:

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

Para obtener más información, consulte las siguientes secciones:

Temas

- Privilegios para MySQL como base de datos de origen
- Conectar a MySQL como origen
- Privilegios para PostgreSQL como base de datos de destino

Privilegios para MySQL como base de datos de origen

A continuación se enumeran los privilegios necesarios para MySQL como origen:

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

Conectar a MySQL como origen

Utilice el siguiente procedimiento para conectarse a su base de datos MySQL de origen con la AWS Schema Conversion Tool.

Para conectarse a una base de datos de MySQL

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Seleccione MySQL y, a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga esas instrucciones:
     1. En AWS Secret, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulte Utilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión a la base de datos de origen de MySQL, siga estas instrucciones:

Parámetro	Acción
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen. Puede conectarse a la base de datos de MySQL de origen a través de un protocolo de direcciones IPv6. Para ello, utilice corchetes para escribir la dirección IP, tal y como se muestra en el siguiente ejemplo. `[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]`
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de su base de datos de origen quede expuesta, la AWS SCT no la almacena por defecto. Si cierra su proyecto de la AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.

Parámetro	Acción
Use SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a su base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Require SSL: seleccione esta opción para conectar con el servidor únicamente a través de SSL. Si selecciona Require SSL, significa que si el servidor no admite SSL, no puede conectar con el servidor. Si no selecciona Require SSL y el servidor no admite SSL, sigue pudiendo conectar con el servidor sin utilizar SSL. Para obtener más información, consulte Configurar MySQL para usar conexiones seguras. • Verify Server Certificate: seleccione esta opción para verificar el certificado del servidor mediante un almacén de confianza. • Trust Store: la ubicación de un almacén de confianza que contenga certificados.
Store Password	La AWS SCT crea un almacén seguro para almacenar certificados SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.
MySql Driver Path	Escriba la ruta al controlador que va a usar para conectarse a la base de datos de origen. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más información, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Privilegios para PostgreSQL como base de datos de destino

Para utilizar PostgreSQL como destino, AWS SCT requiere el privilegio `CREATE ON DATABASE`. Asegúrese de conceder este privilegio a cada base de datos PostgreSQL de destino.

Para usar los sinónimos públicos convertidos, cambie la ruta de búsqueda predeterminada de la base de datos a `"$user", public_synonyms, public`.

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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;
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. A continuación, sustituya *db_name* por el nombre de su base de datos de destino. Por último, sustituya *your_password* por una contraseña segura.

En PostgreSQL, solo el propietario de un esquema o un `superuser` puede anular un esquema. El propietario puede eliminar un esquema y todos los objetos que incluye este esquema, aunque el propietario del esquema no sea propietario de algunos de los objetos.

Si utiliza distintos usuarios para convertir y aplicar diferentes esquemas a la base de datos de destino, es posible que aparezca un mensaje de error cuando AWS SCT no pueda anular un sistema. Para evitar este mensaje de error, utilice el rol de `superuser`.

Usar una base de datos de Oracle como origen para AWS SCT

Puede usar AWS SCT para convertir esquemas, objetos de código de bases de datos y código de aplicación de una base de datos de Oracle a los siguientes destinos:

- Amazon RDS para MySQL

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

Cuando el origen es una base de datos de Oracle, los comentarios pueden convertirse al formato adecuado, por ejemplo, una base de datos PostgreSQL. AWS SCT puede convertir comentarios en tablas, vistas y columnas. Los comentarios pueden incluir apóstrofos; AWS SCT duplica los apóstrofos al convertir instrucciones SQL, del mismo que con los literales de cadena.

Para obtener más información, consulte los siguientes temas.

Temas

- Privilegios para Oracle como origen
- Conexión a Oracle como origen
- Convertir Oracle a Amazon RDS para PostgreSQL o Amazon Aurora (PostgreSQL)
- Convertir Oracle a Amazon RDS para MySQL o Amazon Aurora (MySQL)
- Convertir Oracle a Amazon RDS para Oracle

Privilegios para Oracle como origen

A continuación se enumeran los privilegios necesarios para Oracle como origen:

- CONNECT
- SELECT_CATALOG_ROLE
- SELECT ANY DICTIONARY
- SELECT ON SYS.ARGUMENT$

Conexión a Oracle como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de Oracle de origen con la AWS Schema Conversion Tool.

Para conectarse a una base de datos de Oracle de origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Elija Oracle y, a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:
     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión a la base de datos de origen de Oracle, siga las instrucciones siguientes:

Parámetro	Acción

Tipo	Seleccione el tipo de conexión a la base de datos. En función de su tipo, proporcione la siguiente información adicional:
	• SID
	  • Server name: escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
	  • Server port: el puerto que utilice para conectarse al servidor de su base de datos de origen.
	  • Oracle SID: el ID del sistema (SID) de Oracle. Para encontrar el SID de Oracle, envíe la siguiente consulta a su base de datos de Oracle:

Parámetro	Acción

	`SELECT sys_context('userenv','instance_name') AS SID FROM dual;`
	• Service name
	  • Server name: el nombre DNS o la dirección IP de su servidor de base de datos de origen.
	
	Puede conectarse a la base de datos de Oracle de origen a través de un protocolo de direcciones IPv6. Para ello, utilice corchetes para escribir la dirección IP, tal y como se muestra en el siguiente ejemplo.
	
	`[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]`
	  • Server port: el puerto que utilice para conectarse al servidor de su base de datos de origen.
	  • Service Name: el nombre del servicio de Oracle al que conectarse.
	• TNS Alias
	  • TNS file path: la ruta al archivo que contiene la información de conexión del nombre Transparent Network Substrate (TNS).
	
	Tras elegir el archivo TNS, AWS SCT añade todas las conexiones a las bases de datos de Oracle del archivo a la lista Alias de TNS.
	
	Seleccione esta opción para conectarse a Oracle Real Application Clusters (RAC).
	  • TNS alias: el alias TNS de este archivo que utilizar para conectarse a la base de datos de origen.
	• TNS Connect Identifier
	  • TNS connect identifier: identificador de la información de conexión del TNS registrado.

Parámetro	Acción
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. La primera vez que se conecte a la base de datos de Oracle, debe escribir la ruta al archivo del controlador de Oracle (ojdbc8.jar). Puede descargar el archivo en http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html. Regístrese en el sitio web gratuito de Oracle Technical Network para completar la descarga. AWS SCT utiliza el controlador seleccionado para cualquier conexión futura a la base de datos de Oracle. La ruta del controlador se puede modificar mediante la pestaña Controladores, en Configuración global. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de su base de datos de origen quede expuesta, la AWS SCT no la almacena por defecto. Si cierra su proyecto de la AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Use SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a su base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • SSL authentication: seleccione esta opción para utilizar la autenticación SSL mediante certificado. Configure el almacén de confianza y el almacén de claves en Configuración, Configuración global y Seguridad. • Trust store: el almacén de confianza que se va a utilizar. • Key store: el almacén de claves que se va a utilizar.

Parámetro	Acción
Store Password	AWS SCT crea un almacén seguro para almacenar certificados SSL y contraseñas de la base de datos. Elija esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.
Oracle driver path	Escriba la ruta al controlador que va a usar para conectarse a la base de datos de origen. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más información, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Convertir Oracle a Amazon RDS para PostgreSQL o Amazon Aurora (PostgreSQL)

Al convertir una base de datos de Oracle a RDS para PostgreSQL o Amazon Aurora PostgreSQL, tenga en cuenta lo siguiente.

Temas

- Privilegios para PostgreSQL como base de datos de destino
- Configuración de conversión de Oracle a PostgreSQL
- Conversión de secuencias de Oracle
- Convertir pseudocolumnas ROWID de Oracle
- Conversión de SQL dinámico de Oracle
- Convertir particiones de Oracle

Al convertir objetos del sistema de Oracle a PostgreSQL, AWS SCT realiza las conversiones como se muestra en la tabla siguiente.

Objeto del sistema de Oracle	Descripción	Objeto de PostgreSQL convertido
V$VERSION	Muestra los números de versión de los component es de biblioteca principal en la base de datos de Oracle	aws_oracle_ext.v$v ersion
V$INSTANCE	Una vista que muestra el estado de la instancia actual.	aws_oracle_ext.v$i nstance

Puede utilizarlos AWS SCT para convertir archivos SQL*Plus de Oracle a psql, que es un front-end de PostgreSQL basado en terminales. Para obtener más información, consulte Conversión de SQL de las aplicaciones mediante AWS SCT.

Privilegios para PostgreSQL como base de datos de destino

Para utilizar PostgreSQL como destino, AWS SCT requiere el privilegio `CREATE ON DATABASE`. Asegúrese de conceder este privilegio a cada base de datos PostgreSQL de destino.

Para usar los sinónimos públicos convertidos, cambie la ruta de búsqueda predeterminada de la base de datos a `"$user", public_synonyms, public`.

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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;
```

En el ejemplo anterior, sustituya *`user_name`* por el nombre del usuario. A continuación, sustituya *`db_name`* por el nombre de su base de datos de destino. Por último, sustituya *`your_password`* por una contraseña segura.

Para utilizar Amazon RDS para PostgreSQL como destino, AWS SCT requiere el privilegio `rds_superuser`.

En PostgreSQL, solo el propietario de un esquema o un `superuser` puede anular un esquema. El propietario puede eliminar un esquema y todos los objetos que incluye este esquema, aunque el propietario del esquema no sea propietario de algunos de los objetos.

Si utiliza distintos usuarios para convertir y aplicar diferentes esquemas a la base de datos de destino, es posible que aparezca un mensaje de error cuando AWS SCT no pueda anular un sistema. Para evitar este mensaje de error, utilice el rol de `superuser`.

Configuración de conversión de Oracle a PostgreSQL

Para editar la configuración de conversión de Oracle a PostgreSQL, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Oracle y, a continuación, Oracle – PostgreSQL. AWS SCT muestra todos los ajustes disponibles para la conversión de Oracle a PostgreSQL.

La configuración de conversión de Oracle a PostgreSQL en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.
- Permitir que AWS SCT convierta las vistas materializadas de Oracle en tablas o vistas materializadas en PostgreSQL. En conversión de vistas materializadas como, elija cómo convertir las vistas materializadas de origen.
- Trabajar con el código fuente de Oracle cuando incluya las funciones `TO_CHAR`, `TO_DATE` y `TO_NUMBER` y con parámetros que PostgreSQL no admite. De forma predeterminada, AWS SCT simula el uso de estos parámetros en el código convertido.

  Si el código fuente de Oracle incluye solo los parámetros compatibles con PostgreSQL, puede utilizar funciones nativas `TO_CHAR`, `TO_DATE` y `TO_NUMBER` de PostgreSQL. En este caso, el código convertido funciona más rápido. Para incluir solo estos parámetros, seleccione los siguientes valores:

  - La función TO_CHAR() no utiliza cadenas de formato específicas de Oracle
  - La función TO_DATE() no utiliza cadenas de formato específicas de Oracle
  - La función TO_NUMBER() no utiliza cadenas de formato específicas de Oracle
- Determinar cuándo la base de datos Oracle de origen almacena solo valores enteros en las columnas de clave principal o externa del tipo de datos `NUMBER`, AWS SCT puede convertir estas columnas al tipo de datos `BIGINT`. Este enfoque mejora el rendimiento del código convertido. Para adoptar este enfoque, seleccione Convertir las columnas de clave principal o externa NUMBER en columnas BIGINT. Asegúrese de que el origen no incluya valores de punto flotante en estas columnas para evitar la pérdida de datos.
- Hacer caso omiso a los disparadores y restricciones deshabilitados en el código fuente. Para ello, elija Hacer caso omiso de los disparadores y restricciones deshabilitados.
- Utilizar AWS SCT para convertir variables de cadena denominadas SQL dinámico. El código de la base de datos puede cambiar los valores de estas variables de cadena. Para asegurarse de que AWS SCT siempre convierte el último valor de esta variable de cadena, seleccione Convertir el código SQL dinámico que se crea en las rutinas llamadas.
- Para solucionar ese problema, la versión 10 y anteriores de PostgreSQL no admiten procedimientos. Si usted o sus usuarios no están familiarizados con el uso de procedimientos en PostgreSQL, AWS SCT puede convertir los procedimientos de Oracle en funciones de PostgreSQL. Para ello, seleccione Convertir procedimientos en funciones.
- Ver información adicional sobre los elementos de acción ocurridos. Para ello, puede añadir funciones específicas al paquete de extensión. Para ello, seleccione Agregar un bloqueo de excepciones para los problemas de migración con los siguientes niveles de gravedad. A continuación, elija los niveles de gravedad para aumentar las excepciones definidas por el usuario.
- Trabajar con una base de datos de Oracle de origen que pueda incluir restricciones con los nombres generados automáticamente. Si el código fuente utiliza estos nombres, seleccione Convertir los nombres de restricciones generados por el sistema utilizando los nombres originales del origen. Si el código fuente utiliza estas restricciones pero no sus nombres, desactive esta opción para aumentar la velocidad de conversión.
- Determinar si la base de datos y las aplicaciones se ejecutan en zonas horarias diferentes. De forma predeterminada, AWS SCT simula las zonas horarias del código convertido. Sin embargo, no necesita esta simulación cuando la base de datos y las aplicaciones utilizan la misma zona horaria. En este caso, seleccione La zona horaria del cliente coincide con la zona horaria del servidor.

- Determinar si la base de datos de origen y destino se ejecutan en zonas horarias diferentes. Si es así, la función que simula la función SYSDATE integrada de Oracle devuelve valores diferentes a los de la función de origen. Para asegurarse de que las funciones de origen y destino devuelven los mismos valores, elija Establecer zona horaria predeterminada para la simulación de SYSDATE.
- Usar las funciones de la extensión orafce en el código convertido. Para ello, en Usar implementación orafce, seleccione las funciones que desee utilizar. Para obtener más información sobre orafce, consulte orafce en GitHub.

Conversión de secuencias de Oracle

AWS SCT convierte secuencias de Oracle a PostgreSQL. Si utiliza secuencias para mantener las restricciones de integridad, asegúrese de que los nuevos valores de una secuencia migrada no se superpongan a los valores existentes.

Para rellenar las secuencias convertidas con el último valor de la base de datos de origen

1. Abra su proyecto de AWS SCT con Oracle como origen.
2. Elija Configuración y, a continuación, elija Configuración de conexiones.
3. En la lista superior, elija Oracle y, a continuación, Oracle – PostgreSQL. AWS SCT muestra todos los ajustes disponibles para la conversión de Oracle a PostgreSQL.
4. Seleccione Rellenar las secuencias convertidas con el último valor generado en el lado del origen.
5. Seleccione Aceptar para guardar la configuración y cerrar el cuadro de diálogo Configuración de conversión.

Convertir pseudocolumnas ROWID de Oracle

En una base de datos de Oracle, la pseudocolumna ROWID contiene la dirección de la fila de la tabla. La pseudocolumna ROWID es exclusiva de Oracle. Por ello, AWS SCT convierte la pseudocolumna ROWID en una columna de datos en PostgreSQL. Al utilizar esta conversión, puede conservar la información de la pseudocolumna ROWID.

Al convertir la pseudocolumna ROWID, AWS SCT crea una columna de datos con el tipo de datos `bigint`. Si no hay una clave primaria, AWS SCT establece la columna ROWID como la clave principal. Si hay una clave principal, AWS SCT establece la columna ROWID con una restricción única.

Si el código de la base de datos de origen incluye operaciones con pseudocolumnas ROWID, que no puede ejecutar con un tipo de datos numérico, AWS SCT puede crear una columna de datos con ese tipo de datos `character varying`.

Para crear una columna de datos para la pseudocolumna ROWID de Oracle en un proyecto

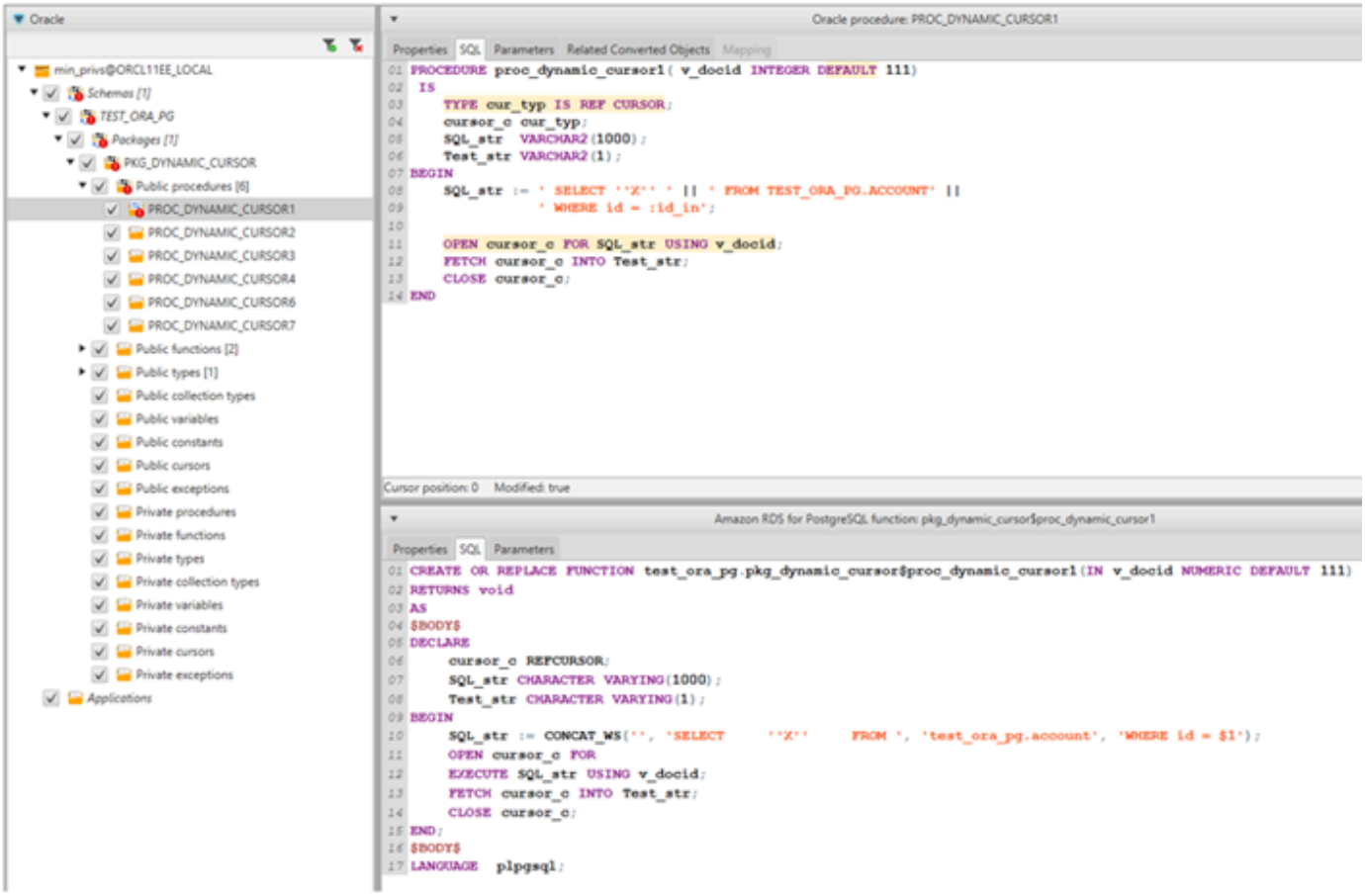
1. Abra su proyecto de AWS SCT con Oracle como origen.
2. Elija Configuración y, a continuación, elija Configuración de conversión.
3. En la lista superior, elija Oracle y, a continuación, Oracle – PostgreSQL. AWS SCT muestra todos los ajustes disponibles para la conversión de Oracle a PostgreSQL.
4. En Generar ID de fila, realice una de las siguientes acciones:
   - Seleccione Generar como identidad para crear una columna de datos numéricos.
   - Seleccione Generar como tipo de dominio de caracteres para crear una columna de datos de caracteres.
5. Seleccione Aceptar para guardar la configuración y cerrar el cuadro de diálogo Configuración de conversión.

Conversión de SQL dinámico de Oracle

Oracle ofrece dos formas de implementar el SQL dinámico: mediante una instrucción EXECUTE IMMEDIATE o llamando a los procedimientos del paquete DBMS_SQL. Si la base de datos de Oracle de origen incluye objetos con SQL dinámico, utilice AWS SCT para convertir instrucciones SQL dinámicas de Oracle a PostgreSQL.

Para convertir instrucciones de SQL dinámico a PostgreSQL

1. Abra su proyecto de AWS SCT con Oracle como origen.
2. Elija un objeto de base de datos que utilice SQL dinámico en la vista en árbol de origen de Oracle.
3. Abra el menú contextual (clic secundario) para el objeto, seleccione Convertir esquema y aceptar reemplazar los objetos si existen. En la siguiente captura de pantalla se muestra el procedimiento convertido bajo el procedimiento de Oracle con SQL dinámico..



Convertir particiones de Oracle

AWS SCT actualmente admite los siguientes métodos de particionamiento:

- Rango
- Enumeración
- Rango de varias columnas
- Hash
- Compuesto (list-list, range-list, list-range, list-hash, range-hash, hash-hash)

Convertir Oracle a Amazon RDS para MySQL o Amazon Aurora (MySQL)

Para simular las funciones de la base de datos de Oracle en código MySQL convertido, utilice el paquete de extensión de Oracle a MySQL en AWS SCT. Para obtener más información acerca de los paquetes de extensión , consulte Uso de paquetes AWS SCT de extensión.

Temas

- Privilegios para MySQL como base de datos de destino
- Configuración de conversión de Oracle a MySQL
- Consideraciones sobre la migración
- Convertir la instrucción WITH en Oracle a Amazon RDS para MySQL o Amazon Aurora MySQL

Privilegios para MySQL como base de datos de destino

Los privilegios necesarios para MySQL como destino se enumeran a continuación:

- CREATE ON *.*
- ALTER ON *.*
- DROP ON *.*
- INDEX ON *.*
- REFERENCES ON *.*
- SELECT ON *.*
- CREATE VIEW ON *.*
- SHOW VIEW ON *.*
- TRIGGER ON *.*
- CREATE ROUTINE ON *.*
- ALTER ROUTINE ON *.*
- EXECUTE ON *.*
- CREATE TEMPORARY TABLES ON *.*
- AWS_LAMBDA_ACCESS
- INSERT, UPDATE ON AWS_ORACLE_EXT.*
- INSERT, UPDATE, DELETE ON AWS_ORACLE_EXT_DATA*.

Si utiliza como destino una base de datos MySQL 5.7 o anterior, conceda el permiso INVOKE LAMBDA *.* en lugar de AWS_LAMBDA_ACCESS. Para las bases de datos MySQL 8.0 y posteriores, conceda el permiso AWS_LAMBDA_ACCESS.

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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';
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. Por último, sustituya *your_password* por una contraseña segura.

Si utiliza como destino una base de datos MySQL 5.7 o anterior, utilice `GRANT INVOKE LAMBDA ON *.* TO 'user_name'` en lugar de `GRANT AWS_LAMBDA_ACCESS TO 'user_name'`.

Para usar Amazon RDS para MySQL o Aurora MySQL como destino, establezca el parámetro `lower_case_table_names` en 1. Este valor significa que el servidor MySQL gestiona los identificadores de nombres de objetos como tablas, índices, disparadores y bases de datos sin distinguir entre mayúsculas y minúsculas. Si ha activado el registro binario en la instancia de destino, establezca el parámetro `log_bin_trust_function_creators` en 1. En este caso, no es necesario utilizar las características `DETERMINISTIC`, `READS SQL DATA` o `NO SQL` para crear funciones almacenadas. Para configurar estos parámetros, cree un grupo de parámetros de base de datos nuevo o modifique uno existente.

Configuración de conversión de Oracle a MySQL

Para editar la configuración de conversión de Oracle a MySQL, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Oracle y, a continuación, Oracle — MySQL. AWS SCT muestra todos los ajustes disponibles para la conversión de Oracle a MySQL.

La configuración de conversión de Oracle a MySQL en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de estos elementos. AWS SCT agrega comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.
- Para solucionarlo, su base de datos Oracle de origen puede utilizar la `ROWID` pseudocolumna, pero MySQL no admite una funcionalidad similar. AWS SCT puede simular la pseudocolumna `ROWID` en el código convertido. Para ello, elija Generar como identidad en ¿Generar ID de fila?.

  Si el código fuente de Oracle no utiliza la pseudocolumna `ROWID`, seleccione No generar para generar en ¿Generar ID de fila? En este caso, el código convertido funciona más rápido.
- Trabajar con el código fuente de Oracle cuando incluye las funciones `TO_CHAR`, `TO_DATE` y `TO_NUMBER` con parámetros que MySQL no admite. De forma predeterminada, AWS SCT simula el uso de estos parámetros en el código convertido.

  Si el código fuente de Oracle incluye solo los parámetros compatibles con PostgreSQL, puede utilizar funciones `TO_CHAR`, `TO_DATE` y `TO_NUMBER` de SQL nativas. En este caso, el código convertido funciona más rápido. Para incluir solo estos parámetros, seleccione los siguientes valores:
  - La función TO_CHAR() no utiliza cadenas de formato específicas de Oracle
  - La función TO_DATE() no utiliza cadenas de formato específicas de Oracle
  - La función TO_NUMBER() no utiliza cadenas de formato específicas de Oracle
- Determinar si la base de datos y las aplicaciones se ejecutan en zonas horarias diferentes. De forma predeterminada, AWS SCT simula las zonas horarias del código convertido. Sin embargo, no necesita esta simulación cuando la base de datos y las aplicaciones utilizan la misma zona horaria. En este caso, seleccione La zona horaria del cliente coincide con la zona horaria del servidor.

Consideraciones sobre la migración

Al convertir Oracle a RDS para MySQL o Aurora MySQL, para cambiar el orden en que se ejecutan las sentencias, puede utilizar una instrucción `GOTO` y una etiqueta. Todas las instrucciones PL/SQL

que van detrás de una instrucción GOTO se omiten y el procesamiento continúa en la etiqueta. Puede usar las instrucciones GOTO y las etiquetas se pueden utilizar en cualquier lugar de un procedimiento, lote o bloque de instrucciones. También puede anidar las instrucciones GOTO.

MySQL no utiliza instrucciones GOTO. Cuando AWS SCT convierte un código que contiene una instrucción GOTO, convierte esta instrucción para utilizar una instrucción BEGIN…END o LOOP…END LOOP.

En la tabla siguiente, puede ver algunos ejemplos de cómo AWS SCT convierte las instrucciones GOTO.

Instrucción de Oracle	Instrucción de MySQL

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

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

Instrucción de Oracle	Instrucción de MySQL
	statement4; .... END

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

Convertir la instrucción WITH en Oracle a Amazon RDS para MySQL o Amazon Aurora MySQL

Puede utilizar la cláusula WITH (subquery_factoring) en Oracle para asignar un nombre (query_name) a un bloque subquery. A continuación, puede hacer referencia al bloque subquery en varios lugares de la consulta especificando el valor de query_name. Si un bloque subquery no contiene enlaces o parámetros (local, procedimiento, función, paquete), entonces AWS SCT convierte la cláusula en una vista o una tabla temporal.

El beneficio de convertir la cláusula en una tabla temporal es que las referencias repetidas a la subconsulta podrían ser más eficientes. La mayor eficiencia se debe a que los datos se recuperan fácilmente de la tabla temporal en lugar de que cada referencia tenga que solicitarlos. Puede emular esto mediante vistas adicionales o una tabla temporal. El nombre de la consulta utiliza el formato `<procedure_name>$<subselect_alias>`.

Puede encontrar ejemplos en la siguiente tabla.

Instrucción de Oracle	Instrucción de MySQL

```
CREATE PROCEDURE
 TEST_ORA_PG.P_WITH_SELECT_V
ARIABLE_01
     (p_state IN NUMBER)
AS
  l_dept_id NUMBER := 1;
BEGIN
FOR cur IN
           (WITH dept_empl(id, name,
 surname,
              lastname, state, dept_id)
              AS
                  (
                    SELECT id, name,
 surname,
                     lastname, state,
 dept_id
                      FROM test_ora_
pg.dept_employees
                     WHERE state =
 p_state AND
                       dept_id =
 l_dept_id)
            SELECT id,state
              FROM dept_empl
            ORDER BY id)  LOOP
  NULL;
END LOOP;
```

```
CREATE PROCEDURE test_ora_pg.P_WITH
_SELECT_VARIABLE_01(IN par_P_STATE
 DOUBLE)
BEGIN
    DECLARE var_l_dept_id DOUBLE
 DEFAULT 1;
    DECLARE var$id VARCHAR (8000);
    DECLARE var$state VARCHAR (8000);
    DECLARE done INT DEFAULT FALSE;
    DECLARE cur CURSOR FOR SELECT
        ID, STATE
        FROM (SELECT
            ID, NAME, SURNAME,
 LASTNAME, STATE, DEPT_ID
            FROM TEST_ORA_PG.DEPT_E
MPLOYEES
            WHERE STATE = par_p_sta
te AND DEPT_ID = var_l_dept_id) AS
 dept_empl
        ORDER BY ID;
    DECLARE CONTINUE HANDLER FOR NOT
 FOUND
        SET done := TRUE;
    OPEN cur;

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

        IF done THEN
            LEAVE read_label;
        END IF;

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

Instrucción de Oracle	Instrucción de MySQL

```
CREATE PROCEDURE
 TEST_ORA_PG.P_WITH_SELECT_R
EGULAR_MULT_01
AS
BEGIN

 FOR cur IN  (
               WITH dept_empl 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_empl
.*,dept.*
                 FROM dept_empl, dept
                 WHERE dept_empl
.dept_id = dept.deptid
              ) LOOP
              NULL;
            END LOOP;
```

```
CREATE VIEW TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$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 = 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$deptname VARCHAR
 (200);
    DECLARE done INT DEFAULT FALSE;
    DECLARE cur CURSOR FOR SELECT
        dept_empl.*, dept.*
        FROM TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept_empl
          ` AS dept_empl,
            TEST_ORA_PG.`P_WIT
H_SELECT_REGULAR_MULT_01$dept
          ` AS dept
```

Instrucción de Oracle	Instrucción de MySQL

	WHERE dept_empl.DEPT_ID = dept.DEPTID;
	DECLARE CONTINUE HANDLER FOR NOT FOUND
	SET done := TRUE;
	OPEN cur;
	
	read_label:
	LOOP
	FETCH cur INTO varID, varNAME, 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_REGULAR_MULT_01()

Instrucción de Oracle	Instrucción de MySQL

```
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;
```

```
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:
```

Instrucción de Oracle	Instrucción de MySQL

	LOOP
	FETCH cur INTO var$ID;
	
	IF done THEN
	LEAVE read_label;
	END IF;
	
	BEGIN
	END;
	END LOOP;
	CLOSE cur;
	END;

Convertir Oracle a Amazon RDS para Oracle

Hay algunos aspectos que deben tenerse en cuenta al migrar un esquema y un código de Oracle a Amazon RDS para Oracle:

- AWS SCT puede añadir objetos de directorio al árbol de objetos. Los objetos de directorio son estructuras lógicas, cada una de las cuales representa un directorio físico del sistema de archivos del servidor. Puede utilizar objetos de directorio con paquetes como DBMS_LOB, UTL_FILE, DBMS_FILE_TRANSFER, la utilidad DATAPUMP, etc.
- AWS SCT permite convertir espacios de tabla de Oracle en una instancia de base de datos de Amazon RDS para Oracle. Oracle almacena los datos de forma lógica en espacios de tabla y físicamente en archivos de datos asociados con el espacio de tabla correspondiente. En Oracle, puede crear un espacio de tabla con nombres de archivos de datos. Amazon RDS admite los archivos administrados de Oracle (OMF) para los archivos de datos, los archivos de registro y los archivos de control. AWS SCT crea los archivos de datos necesarios durante la conversión.
- AWS SCT puede convertir funciones y privilegios del nivel del servidor. El motor de base de datos Oracle utiliza la seguridad basada en roles. Una función es un conjunto de privilegios que se pueden conceder a un usuario o que se pueden revocar. Una función predefinida de Amazon RDS llamada DBA normalmente admite todos los privilegios administrativos de un motor de base de datos de Oracle. Los privilegios siguientes no están disponibles para el rol DBA en una instancia de base de datos de Amazon RDS que utiliza el motor Oracle:
  - Alter database

- Alter system
- Create any directory
- Grant any privilege
- Grant any role
- Create external job

Puede conceder todos los demás privilegios a una función de usuario de Amazon RDS for Oracle, incluidos los de filtrado avanzado y de columnas.

- AWS SCT admite la conversión de trabajos de Oracle en trabajos que pueden ejecutarse en Amazon RDS para Oracle. Existen algunas limitaciones en la conversión, entre las que se incluyen las siguientes:
  - No se admiten los trabajos ejecutables.
  - No se admiten los trabajos de programación que usan el tipo de datos ANYDATA como un argumento.
- Oracle Real Application Clusters (RAC) One Node es una opción de Oracle Database Enterprise Edition que se introdujo con Oracle Database 11g versión 2. Amazon RDS for Oracle no admite la característica RAC. Para obtener una alta disponibilidad, utilice Multi-AZ de Amazon RDS.

  En una implementación multi-AZ, Amazon RDS aprovisiona y mantiene automáticamente una réplica en espera sincrónica dentro de una zona de disponibilidad diferente. La instancia de base de datos principal se replica sincrónicamente en las zonas de disponibilidad en una réplica en espera. Esta funcionalidad proporciona redundancia de datos, elimina los bloqueos de E/S y minimiza los picos de latencia durante los backups del sistema.
- Oracle Spatial proporciona un esquema y funciones SQL que facilitan el almacenamiento, la recuperación, la actualización y la consulta de colecciones de datos espaciales en una base de datos de Oracle. Oracle Locator proporciona capacidades que suelen ser necesarias para admitir aplicaciones basadas en servicio de Internet e inalámbricas y soluciones de Sistemas de información geográfica (SIG) basadas en el partner. Oracle Locator es una subred limitada de Oracle Spatial.

  Para utilizar las características de Oracle Spatial u Oracle Locator, agregue la opción SPATIAL o la opción LOCATOR (mutuamente excluyentes) al grupo de opciones de la instancia de base de datos.

  Existen algunas requisitos previos que debe cumplir para poder usar Oracle Spatial u Oracle Locator en una instancia de base de datos de Amazon RDS for Oracle:

  - La instancia debe usar Oracle Enterprise Edition versión 12.1.0.2.v6 o posterior o la versión 11.2.0.4.v10 o posterior.
  - La instancia debe estar dentro de una nube virtual privada (VPC).
  - La instancia debe usar la clase de instancia de base de datos que es compatible con la característica de Oracle. Por ejemplo, Oracle Spatial no se admite para las clases de instancia de base de datos db.m1.small, db.t1.micro, db.t2.micro o db.t2.small. Para obtener más información, consulte Compatibilidad de clases de instancias de bases de datos con Oracle.
  - La instancia debe tener habilitada la opción Auto Minor Version Upgrade (Actualización automática de versiones menores). Amazon RDS actualiza automáticamente su instancia de base de datos a la versión más reciente de Oracle PSU si hay vulnerabilidades de seguridad con una puntuación de CVSS superior a 9 u otras vulnerabilidades de seguridad anunciadas. Para obtener más información, consulte

    Configuración de las instancias de base de datos de Oracle.
  - Si la versión de su instancia de base de datos es la 11.2.0.4.v10 o posterior, debe instalar la opción XMLDB. Para obtener más información, consulte

    Oracle XML DB.
  - Debe tener una licencia de Oracle Spatial de Oracle. Para obtener más información, consulte Oracle Spatial and Graph en la documentación de Oracle.
- Data Guard se incluye con Oracle Database Enterprise Edition. Para obtener una alta disponibilidad, utilice la característica Multi-AZ de Amazon RDS.

  En una implementación multi-AZ, Amazon RDS aprovisiona y mantiene automáticamente una réplica en espera sincrónica dentro de una zona de disponibilidad diferente. La instancia de base de datos principal se replica sincrónicamente en las zonas de disponibilidad en una réplica en espera. Esta funcionalidad proporciona redundancia de datos, elimina los bloqueos de E/S y minimiza los picos de latencia durante los backups del sistema.
- AWS SCT permite convertir objetos de Oracle DBMS_SCHEDULER al migrar a Amazon RDS para Oracle. El informe de evaluación de AWS SCT indica si un objeto de programación se puede convertir. Para obtener más información acerca del uso de objetos de programación con Amazon RDS, consulte la documentación de Amazon RDS.
- En el caso de conversiones de Oracle a Amazon RDS for Oracle, se admiten los enlaces de base de datos. Un enlace de base de datos es un objeto de esquema en una base de datos que le permite obtener acceso a los objetos de otra base de datos. No es necesario que la otra base

de datos sea de Oracle. Sin embargo, para obtener acceso a las bases de datos que no son de Oracle, debe utilizar Oracle Heterogeneous Services.

Una vez creado un enlace de base de datos, puede utilizar el enlace en instrucciones SQL para hacer referencia a tablas, vistas y objetos PL/SQL en la otra base de datos. Para utilizar un enlace de base de datos, añada `@dblink` al nombre de tabla, vista u objeto PL/SQL. Puede consultar una tabla o una vista de la otra base de datos con la instrucción SELECT. Para obtener más información acerca del uso de enlaces de base de datos de Oracle, consulte la documentación de Oracle.

Para obtener más información acerca del uso de enlaces de base de datos con Amazon RDS, consulte la documentación de Amazon RDS.

- El informe de evaluación de AWS SCT proporciona métricas de servidor para la conversión. Estas métricas acerca de la instancia de Oracle incluyen:
  - Capacidad de memoria y computación de la instancia de base de datos de destino.
  - Características de Oracle, como Real Application Clusters no compatibles que Amazon RDS no admite.
  - Carga de lectura-escritura en disco.
  - Rendimiento de disco medio total
  - Información del servidor, como nombre de servidor, sistema operativo, alojamiento, nombre y conjunto de caracteres.

Privilegios para RDS para Oracle como destino

Para migrar a Amazon RDS para Oracle, cree un usuario de base de datos privilegiado. Puede utilizar el siguiente código de ejemplo.

```
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;
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. Por último, sustituya *your_password* por una contraseña segura.

Limitaciones al convertir de Oracle a Amazon RDS para Oracle

Hay que tener en cuenta algunas limitaciones al migrar un esquema y un código de Oracle a Amazon RDS para Oracle:

- Una función predefinida de Amazon RDS llamada DBA normalmente admite todos los privilegios administrativos de un motor de base de datos de Oracle. Los privilegios siguientes no están disponibles para el rol DBA en una instancia de base de datos de Amazon RDS que utiliza el motor Oracle:

- Alter database
- Alter system
- Create any directory
- Grant any privilege
- Grant any role
- Create external job

  Todos los demás privilegios se pueden conceder a una función de usuario de Oracle RDS.
- Amazon RDS para Oracle admite las auditorías tradicionales y detalladas mediante el paquete DBMS_FGA y Oracle Unified Auditing.
- Amazon RDS para Oracle no admite la captura de datos de cambios (CDC). Para usar la CDC durante y después de la migración de una base de datos, utilice AWS Database Migration Service.

Utilizar PostgreSQL como origen para AWS SCT

Puede usar AWS SCT para convertir esquemas, objetos de código de bases de datos y código de aplicación de PostgreSQL a los siguientes destinos:

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

Para obtener más información, consulte las siguientes secciones:

Temas

- Privilegios para PostgreSQL como base de datos de origen
- Conectar a PostgreSQL como origen
- Privilegios para MySQL como base de datos de destino

Privilegios para PostgreSQL como base de datos de origen

Los privilegios necesarios para PostgreSQL como origen son los siguientes:

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

Conectar a PostgreSQL como origen

Utilice el siguiente procedimiento para conectarse a su base de datos PostgreSQL de origen con la AWS Schema Conversion Tool.

Para conectarse a una base de datos de PostgreSQL de origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Elija PostgreSQL y, a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, lleve a cabo las siguientes instruccioness:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión a la base de datos de origen de PostgreSQL, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen. Puede conectarse a la base de datos de PostgreSQL de origen a través de un protocolo de direcciones IPv6. Para ello, utilice corchetes para escribir la dirección IP, tal y como se muestra en el siguiente ejemplo. `[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]`
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
Base de datos	Escriba el nombre de la base de datos de PostgreSQL.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de su base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de la AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.

Parámetro	Acción
Use SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a su base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Verify Server Certificate: seleccione esta opción para verificar el certificado del servidor mediante un almacén de confianza. • Trust Store: la ubicación de un almacén de confianza que contenga certificados. Para que esta ubicación aparezca en la sección Configuración global, asegúrese de añadirla.
Store Password	La AWS SCT crea un almacén seguro para almacenar certificados SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.
PostgreSQL Driver Path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Privilegios para MySQL como base de datos de destino

Los privilegios necesarios para MySQL como destino se enumeran a continuación:

- CREATE ON *.*
- ALTER ON *.*
- DROP ON *.*
- INDEX ON *.*
- REFERENCES ON *.*
- SELECT ON *.*
- CREATE VIEW ON *.*
- SHOW VIEW ON *.*
- TRIGGER ON *.*
- CREATE ROUTINE ON *.*
- ALTER ROUTINE ON *.*
- EXECUTE ON *.*
- INSERT, UPDATE ON AWS_POSTGRESQL_EXT.*
- INSERT, UPDATE, DELETE ON AWS_POSTGRESQL_EXT_DATA.*
- CREATE TEMPORARY TABLES ON AWS_POSTGRESQL_EXT_DATA.*

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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';
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. Por último, sustituya *your_password* por una contraseña segura.

Para usar Amazon RDS para MySQL o Aurora MySQL como destino, establezca el parámetro `lower_case_table_names` en 1. Este valor significa que el servidor MySQL gestiona los identificadores de nombres de objetos como tablas, índices, disparadores y bases de datos sin distinguir entre mayúsculas y minúsculas. Si ha activado el registro binario en la instancia de destino, establezca el parámetro `log_bin_trust_function_creators` en 1. En este caso, no es necesario utilizar las características `DETERMINISTIC`, `READS SQL DATA` o `NO SQL` para crear funciones almacenadas. Para configurar estos parámetros, cree un grupo de parámetros de base de datos nuevo o modifique uno existente.

Uso de SAP ASE (Sybase ASE) como origen para AWS SCT

Puede usar AWS SCT para convertir esquemas, objetos de código de bases de datos y código de aplicación de SAP (Sybase) Adaptive Server Enterprise (ASE) a los siguientes destinos:

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

Para obtener más información, consulte las siguientes secciones:

Temas

- Privilegios para SAP ASE como base de datos de origen
- Conectar a SAP ASE (Sybase) como origen
- Privilegios para MySQL como base de datos de destino
- Configuración de conversión de SAP ASE a MySQL
- Privilegios para PostgreSQL como base de datos de destino

- Configuración de conversión de SAP ASE a PostgreSQL

Privilegios para SAP ASE como base de datos de origen

Para utilizar una base de datos de SAP ASE como origen, debe crear un usuario de base de datos y conceder permisos. Para ello, siga estos pasos.

Cree y configure un usuario de base de datos

1. Conéctese a la base de datos de origen.
2. Cree un usuario de base de datos con los siguientes comandos. Proporcione una contraseña para el usuario nuevo.

```
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. Para cada base de datos que vaya a migrar, otorgue los siguientes privilegios.

```
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
```

Conectar a SAP ASE (Sybase) como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de origen de SAP ASE con AWS Schema Conversion Tool.

Para conectarse a una base de datos de SAP ASE

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Elija SAP ASE, y a continuación, Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión a la base de datos de origen de SAP ASE, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
Base de datos	Escriba el nombre de la base de datos de SAP ASE.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen.

Parámetro	Acción
	Note AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de su base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de la AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Use SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a su base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Verify Server Certificate: seleccione esta opción para verificar el certificado del servidor mediante un almacén de confianza. • Trust Store: la ubicación de un almacén de confianza que contenga certificados.
Store Password	La AWS SCT crea un almacén seguro para almacenar certificados SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.

Parámetro	Acción
SAP ASE driver path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Privilegios para MySQL como base de datos de destino

Los privilegios necesarios para MySQL como destino se enumeran a continuación:

- CREATE ON *.*
- ALTER ON *.*
- DROP ON *.*
- INDEX ON *.*
- REFERENCES ON *.*
- SELECT ON *.*
- CREATE VIEW ON *.*
- SHOW VIEW ON *.*
- TRIGGER ON *.*
- CREATE ROUTINE ON *.*
- ALTER ROUTINE ON *.*
- EXECUTE ON *.*
- INSERT, UPDATE ON AWS_SAPASE_EXT*.

- CREATE TEMPORARY TABLES ON AWS_SAPASE_EXT*.

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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';
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. Por último, sustituya *your_password* por una contraseña segura.

Para usar Amazon RDS para MySQL o Aurora MySQL como destino, establezca el parámetro `lower_case_table_names` en 1. Este valor significa que el servidor MySQL gestiona los identificadores de nombres de objetos como tablas, índices, disparadores y bases de datos sin distinguir entre mayúsculas y minúsculas. Si ha activado el registro binario en la instancia de destino, establezca el parámetro `log_bin_trust_function_creators` en 1. En este caso, no es necesario utilizar las características `DETERMINISTIC`, `READS SQL DATA` o `NO SQL` para crear funciones almacenadas. Para configurar estos parámetros, cree un grupo de parámetros de base de datos nuevo o modifique uno existente.

Configuración de conversión de SAP ASE a MySQL

Para editar la configuración de conversión de SAP ASE a MySQL, seleccione Configuración y, a continuación, elija Configuración de conversión. En la lista superior, elija SAP ASE y, a continuación, SAP ASE – MySQL o SAP ASE – Amazon Aurora (compatible con MySQL). AWS SCT muestra todas las configuraciones disponibles para la conversión de SAP ASE a PostgreSQL.

La configuración de conversión de SAP ASE a MySQL en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Agregar comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.
- Usar los nombres exactos de los objetos de la base de datos de origen en el código convertido.

  De forma predeterminada, AWS SCT convierte los nombres de los objetos, variables y parámetros de la base de datos a minúsculas. Para mantener las mayúsculas y minúsculas originales de estos nombres, seleccione Distinguir entre mayúsculas y minúsculas en los nombres de los objetos de la base de datos de origen. Elija esta opción si utiliza nombres de objetos que distingan mayúsculas de minúsculas en el servidor de base de datos de SAP ASE de origen.

Privilegios para PostgreSQL como base de datos de destino

Para utilizar PostgreSQL como destino, AWS SCT requiere el privilegio `CREATE ON DATABASE`. Asegúrese de conceder este privilegio a cada base de datos PostgreSQL de destino.

Para usar los sinónimos públicos convertidos, cambie la ruta de búsqueda predeterminada de la base de datos a `"$user", public_synonyms, public`.

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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;
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. A continuación, sustituya *db_name* por el nombre de su base de datos de destino. Por último, sustituya *your_password* por una contraseña segura.

En PostgreSQL, solo el propietario de un esquema o un `superuser` puede anular un esquema. El propietario puede eliminar un esquema y todos los objetos que incluye este esquema, aunque el propietario del esquema no sea propietario de algunos de los objetos.

Si utiliza distintos usuarios para convertir y aplicar diferentes esquemas a la base de datos de destino, es posible que aparezca un mensaje de error cuando AWS SCT no pueda anular un sistema. Para evitar este mensaje de error, utilice el rol de `superuser`.

Configuración de conversión de SAP ASE a PostgreSQL

Para editar la configuración de conversión de SAP ASE a PostgreSQL, seleccione Configuración y, a continuación, elija Configuración de conversión. En la lista superior, elija SAP ASE y, a continuación, elija SAP ASE – PostgreSQL o SAP ASE – Amazon Aurora (compatible con PostgreSQL). AWS SCT muestra todas las configuraciones disponibles para la conversión de SAP ASE a PostgreSQL.

La configuración de conversión de SAP ASE a PostgreSQL en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Agregar comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.
- Definir la plantilla que se utilizará para los nombres de los esquemas del código convertido. En Plantilla de generación de nombres de esquema, elija una de las siguientes opciones:
  - **<source_db>**: utiliza el nombre de la base de datos de SAP ASE como nombre de esquema en PostgreSQL.
  - **<source_schema>**: utiliza el nombre del esquema de SAP ASE como nombre de esquema en PostgreSQL.
  - **<source_db>_<schema>**: utiliza una combinación de los nombres de la base de datos y del esquema de SAP ASE como nombre de esquema en PostgreSQL.
- Usar los nombres exactos de los objetos de la base de datos de origen en el código convertido.

De forma predeterminada, AWS SCT convierte los nombres de los objetos, variables y parámetros de la base de datos a minúsculas. Para mantener las mayúsculas y minúsculas originales de estos nombres, seleccione Distinguir entre mayúsculas y minúsculas en los nombres de los objetos de la base de datos de origen. Elija esta opción si utiliza nombres de objetos que distingan mayúsculas de minúsculas en el servidor de base de datos de SAP ASE de origen.

Para las operaciones que distinguen entre mayúsculas y minúsculas, AWS SCT puede evitar la conversión de los nombres de los objetos de la base de datos a minúsculas. Para ello, seleccione Evitar la conversión a minúsculas para las operaciones que distingan entre mayúsculas y minúsculas.

- Permitir el uso de índices con el mismo nombre en diferentes tablas de SAP ASE.

  En PostgreSQL, todos los nombres de índice que utilice en el esquema deben ser únicos. Para asegurarse de que AWS SCT genera nombres únicos para todos los índices, seleccione Generar nombres únicos para los índices.

Uso de Microsoft SQL Server como origen para AWS SCT

Puede usar AWS SCT para convertir esquemas, objetos de código de bases de datos y código de aplicación de SQL Server a los siguientes destinos:

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

> **Note**
>
> AWS SCT no admite el uso de Amazon RDS para SQL Server como origen.

Puede utilizar AWS SCT para crear un informe de evaluación para la migración de esquemas, objetos de código de base de datos y código de aplicación de SQL Server a Babelfish para Aurora PostgreSQL, tal y como se describe a continuación.

Temas

- Privilegios para Microsoft SQL Server como origen
- Uso de la autenticación de Windows al utilizar Microsoft SQL Server como origen
- Conectar a SQL Server como origen
- Conversión de SQL Server a MySQL
- Convertir SQL Server a PostgreSQL
- Convertir SQL Server a Amazon RDS para SQL Server

Privilegios para Microsoft SQL Server como origen

Los privilegios necesarios para Microsoft SQL Server como origen son los siguientes:

- VIEW DEFINITION
- VIEW DATABASE STATE

El privilegio `VIEW DEFINITION` permite a los usuarios que tienen acceso público ver las definiciones de objetos. AWS SCT utiliza el privilegio `VIEW DATABASE STATE` para comprobar las características de la edición SQL Server Enterprise.

Repita la concesión para cada base de datos cuyo esquema vaya a convertir.

Además, conceda los siguientes privilegios en la base de datos `master`:

- VIEW SERVER STATE
- VIEW ANY DEFINITION

AWS SCT utiliza el privilegio `VIEW SERVER STATE` para recopilar los ajustes y la configuración del servidor. Asegúrese de conceder el privilegio `VIEW ANY DEFINITION` para ver los puntos de conexión.

Para leer información sobre Microsoft Analysis Services, ejecute el siguiente comando en la base de datos `master`.

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

En el ejemplo anterior, sustituya el marcador de posición *<user_name>* por el nombre del usuario al que concedió anteriormente los privilegios.

Para leer información sobre el Agente SQL Server, agregue su usuario al rol `SQLAgentUser`. Ejecute el siguiente comando en la base de datos `msdb`.

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

En el ejemplo anterior, sustituya el marcador de posición *<SQLAgentRole>* por el nombre del rol del Agente SQL Server. A continuación, sustituya el marcador de posición *<user_name>* por el nombre del usuario al que concedió anteriormente los privilegios. Para obtener más información, consulte Agregar un usuario al rol de SQLAgentUser en la Guía del usuario de Amazon RDS.

Para detectar el envío de registros, conceda el privilegio `SELECT on dbo.log_shipping_primary_databases` en la base de datos de `msdb`.

Para utilizar el enfoque de notificación de la replicación del DDL, conceda el privilegio `RECEIVE ON` *<schema_name>*`.`*<queue_name>* en las bases de datos de origen. En este ejemplo, sustituya el marcador de posición *<schema_name>* por el nombre del esquema de la base de datos. A continuación, sustituya el marcador de posición *<queue_name>* por el nombre de una tabla de colas.

Uso de la autenticación de Windows al utilizar Microsoft SQL Server como origen

Si la aplicación se ejecuta en una intranet basada en Windows, es posible que pueda utilizar la autenticación de Windows para el acceso a las bases de datos. La autenticación de Windows utiliza la identidad de Windows actual establecida en el subproceso de sistema operativo para obtener acceso a la base de datos de SQL Server. A continuación, puede mapear la identidad de Windows a una base de datos de SQL Server y sus permisos. Para conectarse a SQL Server con la autenticación de Windows, debe especificar la identidad de Windows que utiliza la aplicación. También debe conceder a la identidad de Windows acceso a la base de datos de SQL Server.

SQL Server tiene dos modos de acceso: autenticación de Windows y mixto. El modo de autenticación de Windows habilita dicha autenticación y deshabilita la de SQL Server. El modo mixto habilita tanto la autenticación de Windows como la de SQL Server. La autenticación de Windows

siempre está disponible y no se puede deshabilitar. Para obtener más información acerca de la autenticación de Windows, consulte la documentación de Microsoft Windows.

El ejemplo posible para crear un usuario en TEST_DB se muestra a continuación.

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

Usar la autenticación de Windows con una conexión JDBC

El controlador JDBC no admite la autenticación de Windows si dicho controlador se utiliza en sistemas operativos que no son de Windows. Las credenciales de autenticación de Windows, como el nombre de usuario y la contraseña, no se especifican de forma automática al conectarse a SQL Server desde sistemas operativos que no son de Windows. En estos casos, las aplicaciones deben utilizar la autenticación de SQL Server.

En la cadena de conexión de JDBC, debe especificarse el parámetro `integratedSecurity` para conectarse mediante la autenticación de Windows. El controlador JDBC admite la autenticación de Windows integrada para los sistemas operativos de Windows a través del parámetro de cadena de conexión `integratedSecurity`.

Para usar la autenticación integrada

1. Instale el controlador JDBC.
2. Copie el archivo `sqljdbc_auth.dll` en un directorio de la ruta de sistema de Windows en el equipo donde está instalado el controlador JDBC.

   Los archivos `sqljdbc_auth.dll` se instalan en la ubicación siguiente:

   <directorio de instalación>\sqljdbc_<versión>\<idioma>\auth\

Cuando intenta establecer una conexión con una base de datos de SQL Server con la autenticación de Windows, es posible que aparezca este error: Este controlador no está configurado para la autenticación integrada. Este problema se puede resolver con las siguientes acciones:

- Declare dos variables que apunten a la ruta instalada de JDBC:

`variable name: SQLJDBC_HOME; variable value: D:\lib\JDBC4.1\enu` (donde reside su archivo sqljdbc4.jar);

`variable name: SQLJDBC_AUTH_HOME; variable value: D\lib\JDBC4.1\enu\auth\x86` (si está ejecutando un sistema operativo de 32 bits) o `D\lib\JDBC4.1\enu\auth\x64` (si está ejecutando un sistema operativo de 64 bits). Aquí es donde reside su archivo `sqljdbc_auth.dll`.

- Copie `sqljdbc_auth.dll` en la carpeta donde se está ejecutando JDK/JRE. Puede copiarlo en la carpeta lib, la carpeta bin, etc. Como ejemplo, es posible que copie en la siguiente carpeta.

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

- Asegúrese de que en la carpeta de bibliotecas JDBC,solo tiene el archivo SQLJDBC4.jar. Quite otros archivos sqljdbc*.jar de esa carpeta (o cópielos en otra carpeta). Si va a agregar el controlador como parte de su programa, asegúrese de que agrega solo SQLJDBC4.jar como el controlador que se utilizará.
- Copie el archivo sqljdbc_auth.dll en la carpeta con su aplicación.

> **Note**
>
> Si ejecuta una máquina virtual de Java (JVM) de 32 bits, utilice el archivo sqljdbc_auth.dll de la carpeta x86, incluso si la versión del sistema operativo es x64. Si ejecuta una JVM de 64 bits en un procesador x64, utilice el archivo sqljdbc_auth.dll de la carpeta x64.

Al conectarse a una base de datos de SQL Server, puede elegir Autenticación de Windows o Autenticación de SQL Server en la opción Autenticación.

Conectar a SQL Server como origen

Utilice el siguiente procedimiento para conectarse a su base de datos Microsoft SQL Server de origen con la AWS Schema Conversion Tool.

Para conectarse a una base de datos de Microsoft SQL Server de origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Elija Microsoft SQL Server y, a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión a la base de datos de origen de Microsoft SQL Server, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del servicio de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen. Puede conectarse a la base de datos de SQL Server de origen a través de un protocolo de direcciones IPv6. Para ello, utilice corchetes para escribir la dirección IP, tal y como se muestra en el siguiente ejemplo. `[2001:db8:ffff:ffff:ffff:ffff:ffff:fffe]`

Parámetro	Acción
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
Instance name	Escriba el nombre de la instancia de la base de datos de SQL Server. Para encontrar el nombre de la instancia, ejecute la consulta `SELECT @@servername;` en su base de datos de SQL Server.
Autenticación	Elija el tipo de autenticación entre Autenticación de Windows y Autenticación de SQL Server.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de su base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de la AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Use SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a su base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Trust Server Certificate: seleccione esta opción para confiar en el certificado del servidor. • Trust Store: la ubicación de un almacén de confianza que contenga certificados. Para que esta ubicación aparezca en la sección Configuración global, asegúrese de agregarla.

Parámetro	Acción
Store Password	La AWS SCT crea un almacén seguro para almacenar certificados SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.
Sql Server Driver Path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.
Windows Authentication library	Introduzca la ruta del archivo `sqljdbc_auth.dll` . De forma predeterminada, este archivo se instala en la siguiente ubicación: `<installation directory of the JDBC driver>sqljdbc_<version> \<language> \auth\`

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Conversión de SQL Server a MySQL

Para simular las funciones de la base de datos de Microsoft SQL Server en código MySQL convertido, utilice el paquete de extensión de SQL Server a MySQL en AWS SCT. Para obtener más información acerca de los paquetes de extensión, consulte Uso de paquetes AWS SCT de extensión.

Temas

- Privilegios para MySQL como base de datos de destino

- Configuración de conversión de SQL Server a MySQL
- Consideraciones sobre la migración

Privilegios para MySQL como base de datos de destino

Los privilegios necesarios para MySQL como destino se enumeran a continuación:

- CREATE ON *.*
- ALTER ON *.*
- DROP ON *.*
- INDEX ON *.*
- REFERENCES ON *.*
- SELECT ON *.*
- CREATE VIEW ON *.*
- SHOW VIEW ON *.*
- TRIGGER ON *.*
- CREATE ROUTINE ON *.*
- ALTER ROUTINE ON *.*
- EXECUTE ON *.*
- INSERT, UPDATE ON AWS_SQLSERVER_EXT.*
- INSERT, UPDATE, DELETE ON AWS_SQLSERVER_EXT_DATA*.
- CREATE TEMPORARY TABLES ON AWS_SQLSERVER_EXT_DATA*.

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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';
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. Por último, sustituya *your_password* por una contraseña segura.

Si utiliza como destino una base de datos MySQL 5.7 o anterior, ejecute el siguiente comando. Para las bases de datos MySQL versión 8.0 y posterior, este comando está obsoleto.

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

Para usar Amazon RDS para MySQL o Aurora MySQL como destino, establezca el parámetro `lower_case_table_names` en 1. Este valor significa que el servidor MySQL gestiona los identificadores de nombres de objetos como tablas, índices, disparadores y bases de datos sin distinguir entre mayúsculas y minúsculas. Si ha activado el registro binario en la instancia de destino, establezca el parámetro `log_bin_trust_function_creators` en 1. En este caso, no es necesario utilizar las características `DETERMINISTIC`, `READS SQL DATA` o `NO SQL` para crear funciones almacenadas. Para configurar estos parámetros, cree un grupo de parámetros de base de datos nuevo o modifique uno existente.

Configuración de conversión de SQL Server a MySQL

Para editar la configuración de conversión de SQL Server a MySQL, en AWS SCT, seleccion Configuración y, a continuación, elija Configuración de conversión. En la lista superior, elija SQL Server y, a continuación, SQL Server — MySQL. AWS SCT muestra todos los ajustes disponibles para la conversión de SQL Server a MySQL.

La configuración de conversión de SQL Server a MySQL en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTagrega

comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.

- Permitir que la base de datos de SQL Server de origen almacene la salida de `EXEC` en una tabla. AWS SCT crea tablas temporales y un procedimiento adicional para simular esta característica. Para usar esta simulación, seleccione Crear rutinas adicionales para gestionar conjuntos de datos abiertos.

Consideraciones sobre la migración

Tenga en cuenta estos aspectos al migrar un esquema de SQL Server a MySQL:

- MySQL no admite la instrucción `MERGE`. Sin embargo, AWS SCT puede simular la instrucción `MERGE` durante la conversión utilizando la cláusula `INSERT ON DUPLICATE KEY` y las instrucciones `UPDATE FROM and DELETE FROM`.

  Para conseguir una simulación correcta con `INSERT ON DUPLICATE KEY`, asegúrese de que existe una restricción o una clave principal únicas en la base de datos MySQL de destino.
- Puede usar una instrucción `GOTO` y una etiqueta para cambiar el orden en el que se ejecutan las instrucciones. Todas las instrucciones Transact-SQL que van detrás de una instrucción `GOTO` se omiten y el procesamiento continúa en la etiqueta. Puede usar las instrucciones `GOTO` y las etiquetas se pueden utilizar en cualquier lugar de un procedimiento, lote o bloque de instrucciones. También puede anidar instrucciones `GOTO`.

  MySQL no utiliza instrucciones `GOTO`. Cuando AWS SCT convierte un código que contiene una instrucción `GOTO`, convierte esta instrucción para utilizar una instrucción `BEGIN…END` o `LOOP… END LOOP`. En la tabla siguiente, puede ver algunos ejemplos de cómo AWS SCT convierte las instrucciones `GOTO`.

Instrucción de SQL Server	Instrucción de MySQL

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

Instrucción de SQL Server	Instrucción de MySQL
.... GOTO label1; statement2; .... label1: Statement3; .... END	.... statement1; .... LEAVE label1; statement2; .... END; Statement3; .... END
BEGIN .... statement1; .... label1: statement2; .... GOTO label1; statement3; .... statement4; .... END	BEGIN .... statement1; .... label1: LOOP statement2; .... ITERATE label1; LEAVE label1; END LOOP; statement3; .... statement4; .... END

Instrucción de SQL Server	Instrucción de MySQL

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

- MySQL no admite funciones con valores de tabla de varias instrucciones. AWS SCT simula funciones con valores de tabla durante una conversión creando tablas temporales y reescribiendo instrucciones para utilizar estas tablas temporales.

Convertir SQL Server a PostgreSQL

Puede usar el paquete de extensión de SQL Server a PostgreSQL en AWS SCT. Este paquete de extensión simula las funciones de la base de datos de SQL Server en el código PostgreSQL convertido. Utilice el paquete de extensión de SQL Server a PostgreSQL para simular las funcionalidades de Agente SQL Server y correo electrónico de base de datos de SQL Server. Para obtener más información acerca de los paquetes de extensión, consulte Uso de paquetes AWS SCT de extensión.

Temas

- Privilegios para PostgreSQL como base de datos de destino
- Conversión de SQL Server a PostgreSQL
- Convertir particiones de SQL Server en particiones de PostgreSQL versión 10
- Consideraciones sobre la migración
- Usar un paquete de extensión de AWS SCT para simular el Agente SQL Server en PostgreSQL

- Usar un paquete de extensión de AWS SCT para simular el correo electrónico de base de datos de SQL Server en PostgreSQL

Privilegios para PostgreSQL como base de datos de destino

Para utilizar PostgreSQL como destino, AWS SCT requiere el privilegio `CREATE ON DATABASE`. Asegúrese de conceder este privilegio a cada base de datos PostgreSQL de destino.

Para usar los sinónimos públicos convertidos, cambie la ruta de búsqueda predeterminada de la base de datos a `"$user", public_synonyms, public`.

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los privilegios.

```
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;
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. A continuación, sustituya *db_name* por el nombre de su base de datos de destino. Por último, sustituya *your_password* por una contraseña segura.

En PostgreSQL, solo el propietario de un esquema o un `superuser` puede anular un esquema. El propietario puede eliminar un esquema y todos los objetos que incluye este esquema, aunque el propietario del esquema no sea propietario de algunos de los objetos.

Si utiliza distintos usuarios para convertir y aplicar diferentes esquemas a la base de datos de destino, es posible que aparezca un mensaje de error cuando AWS SCT no pueda anular un sistema. Para evitar este mensaje de error, utilice el rol de `superuser`.

Conversión de SQL Server a PostgreSQL

Para editar la configuración de conversión de SAP ASE a PostgreSQL, seleccione Configuración y, a continuación, elija Configuración de conversión. En la lista superior, elija SQL Server y, a continuación, SQL Server — PostgreSQL. AWS SCT muestra todos los ajustes disponibles para la conversión de SQL Server a PostgreSQL.

La configuración de conversión de SQL Server a PostgreSQL en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Agregar comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTagrega comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.

- Permitir el uso de índices con el mismo nombre en diferentes tablas de SQL Server.

  En PostgreSQL, todos los nombres de índice que utilice en el esquema deben ser únicos. Para asegurarse de que AWS SCT genera nombres únicos para todos los índices, seleccione Generar nombres únicos para los índices.

- Convertir los procedimientos de SQL Server en funciones de PostgreSQL.

  La versión 10 y anteriores de PostgreSQL no admiten procedimientos. Para los usuarios que no están familiarizados con el uso de procedimientos en PostgreSQL, AWS SCT puede convertir los procedimientos a funciones. Para ello, seleccione Convertir procedimientos en funciones.

- Simular la salida de `EXEC` en una tabla.

  Su base de datos de SQL Server de origen puede almacenar la salida de `EXEC` en una tabla. AWS SCT crea tablas temporales y un procedimiento adicional para simular esta característica. Para usar esta simulación, seleccione Crear rutinas adicionales para gestionar conjuntos de datos abiertos.

- Definir la plantilla que se utilizará para los nombres de los esquemas del código convertido. En Plantilla de generación de nombres de esquema, elija una de las siguientes opciones:
  - **<**source_db**>**: utiliza el nombre de la base de datos de SQL Server como nombre de esquema en PostgreSQL.
  - **<**source_schema**>**: utiliza el nombre del esquema de SQL Server como nombre de esquema en PostgreSQL.
  - **<**source_db**>**_**<**schema**>**: utiliza una combinación de los nombres de la base de datos y del esquema de SQL Server como nombre de esquema en PostgreSQL.
- Mantener las mayúsculas y minúsculas de los nombres de los objetos de origen.

Para evitar la conversión de los nombres de los objetos a minúsculas, seleccione Evitar la conversión a minúsculas para las operaciones que distingan mayúsculas y minúsculas. Esta opción solo se aplica cuando se activa la opción de distinguir entre mayúsculas y minúsculas en la base de datos de destino.

- Conservar los nombres de los parámetros de la base de datos de origen.

  Para agregar comillas dobles a los nombres de los parámetros del código convertido, seleccione Conservar los nombres de los parámetros originales.

Convertir particiones de SQL Server en particiones de PostgreSQL versión 10

Al convertir una base de datos de Microsoft SQL Server en Amazon Aurora PostgreSQL Compatible Edition (Aurora PostgreSQL) o Amazon Relational Database Service para PostgreSQL (Amazon RDS para PostgreSQL), tenga en cuenta lo siguiente.

En SQL Server, debe crear particiones con funciones de partición. Cuando una tabla particionada de SQL Server se convierte en una tabla particionada de PostgreSQL versión 10, debe tenerse en cuenta que pueden producirse algunos problemas:

- SQL Server le permite particionar una tabla utilizando una columna sin la restricción NOT NULL. En ese caso, todos los valores NULL van a la partición situada más a la izquierda. PostgreSQL no admite valores NULL con particiones RANGE.
- SQL Server le permite crear claves principales y claves únicas para tablas particionadas. En el caso de PostgreSQL, las claves primarias y las claves únicas se crean directamente para cada partición. Por tanto, la restricción PRIMARY o UNIQUE KEY debe eliminarse de la tabla principal al migrar a PostgreSQL. Los nombres de clave resultantes tienen el formato `<original_key_name>_<partition_number>`.
- SQL Server permite crear restricciones de clave externa que tienen como origen o destino tablas particionadas. PostgreSQL no admite las claves externas que hacen referencia a tablas particionadas. Además, PostgreSQL tampoco admite las referencias de clave externa entre una tabla particionada y otra tabla.
- SQL Server permite crear índices para las tablas particionadas. En PostgreSQL, los índices deben crearse directamente con cada partición. Por lo tanto, los índices deben eliminarse de sus tablas principales al migrar a PostgreSQL. Los nombres de índice resultantes tienen el formato `<original_index_name>_<partition_number>`.
- PostgreSQL no admite índices particionados.

Consideraciones sobre la migración

Hay algunos aspectos que deben tenerse en cuenta al migrar un esquema de SQL Server a PostgreSQL:

- En PostgreSQL, todos los nombres de objeto de un esquema deben ser único, incluidos los índices. Los nombres de los índices deben ser únicos en el esquema de la tabla base. En SQL Server, un nombre de índice puede ser igual en diferentes tablas.

  Para garantizar la exclusividad de los índices, AWS SCT brinda la opción de generar nombres de índice únicos si dichos nombres de índice no lo son. Para ello, seleccione la opción Generate unique index names (Generar nombres de índice únicos) en las propiedades del proyecto. Esta opción está habilitada de forma predeterminada. Si esta opción está habilitada, se crean nombres de índice único con el formato IX_nombre_tabla_nombre_índice. Si esta opción está deshabilitada, los nombres de índice no se modifican.
- Para cambiar el orden en el que se ejecutan las instrucciones, se puede utilizar una instrucción GOTO y una etiqueta. Todas las instrucciones Transact-SQL que van detrás de una instrucción GOTO se omiten y el procesamiento continúa en la etiqueta. Las instrucciones GOTO y las etiquetas se pueden utilizar en cualquier lugar de un procedimiento, lote o bloque de instrucciones. Las instrucciones GOTO también se pueden anidar.

  PostgreSQL no utiliza instrucciones GOTO. Cuando AWS SCT convierte un código que contiene una instrucción GOTO, convierte esta instrucción para que utilizar en su lugar una instrucción BEGIN...END o LOOP...END LOOP. En la tabla siguiente, puede ver algunos ejemplos de cómo AWS SCT convierte las instrucciones GOTO.

  Instrucciones GOTO de SQL Server e instrucciones PostgreSQL convertidas

Instrucción de SQL Server	Instrucción de PostgreSQL

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

Instrucción de SQL Server	Instrucción de PostgreSQL
END	Statement3; .... END
BEGIN .... statement1; .... label1: statement2; .... GOTO label1; statement3; .... statement4; .... END	BEGIN .... statement1; .... label1: LOOP statement2; .... CONTINUE label1; EXIT label1; END LOOP; statement3; .... statement4; .... END

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

- PostgreSQL no admite la instrucción MERGE. AWS SCT simula el comportamiento de una instrucción MERGE de las siguientes maneras:
  - Mediante la construcción INSERT ON CONFLICT.

  - Mediante la instrucción UPDATE FROM DML, como MERGE sin la cláusula WHEN NOT MATCHED.
  - Mediante el uso de CURSOR, como con una cláusula MERGE con DELETE o mediante el uso de una instrucción de condición compleja MERGE ON.
- AWS SCT puede agregar disparadores de bases de datos al árbol de objetos cuando el destino es Amazon RDS.
- AWS SCT puede agregar disparadores de nivel de servidor al árbol de objetos cuando el destino es Amazon RDS.
- SQL Server crea y administra tablas `deleted` y `inserted` de forma automática. Puede utilizar estas tablas temporales que residen en la memoria para probar los efectos de determinadas modificaciones en los datos y establecer las condiciones para las acciones que desencadenan el DML. AWS SCT puede convertir el uso de estas tablas en instrucciones de activación de DML.
- AWS SCT puede agregar servidores enlazados al árbol de objetos cuando el destino es Amazon RDS.
- Al migrar desde Microsoft SQL Server a PostgreSQL, la función SUSER_SNAME integrada se convierte tal y como se indica a continuación:
  - SUSER_SNAME: devuelve el nombre de inicio de sesión asociado a un número de identificación de seguridad (SID).
  - SUSER_SNAME(<sid_usuario_servidor>): no admitido.
  - SUSER_SNAME() CURRENT_USER: devuelve el nombre de usuario del contexto de ejecución actual.
  - SUSER_SNAME (NULL): devuelve NULL.
- Se permite la conversión de funciones con valores de tabla. Las funciones con valores de tabla devuelven una tabla y pueden tomar el lugar de una tabla en una consulta.
- PATINDEX devuelve la posición inicial de la primera coincidencia de un patrón en una expresión especificada en todos los tipos de datos de texto y caracteres válidos. Devuelve ceros si no se encuentra el patrón. Al convertir de SQL Server a Amazon RDS para PostgreSQL, AWS SCT reemplaza el código de aplicación que usa PATINDEX por aws_sqlserver_ext.patindex(<pattern character>, <expression character varying>) .
- En SQL Server, un tipo de tabla definido por el usuario es un tipo que representa la definición de la estructura de una tabla. Se utiliza un tipo de tabla definido por el usuario para declarar parámetros de valor de tabla para procedimientos o funciones almacenados. También puede utilizar un tipo de tabla definido por el usuario para declarar las variables de tabla que desea utilizar en un lote o en

el cuerpo de un procedimiento o función almacenados. AWS SCT simula este tipo en PostgreSQL creando una tabla temporal.

Al convertir de SQL Server a PostgreSQL, AWS SCT convierte los objetos de sistema de SQL Server en objetos reconocibles en PostgreSQL. En la siguiente tabla se muestra cómo se convierten los objetos del sistema.

Casos de uso de MS SQL Server	Sustitución de PostgreSQL
SYS.SCHEMAS	AWS_SQLSERVER_EXT.SYS_SCHEMAS
SYS.TABLES	AWS_SQLSERVER_EXT.SYS_TABLES
SYS.VIEWS	AWS_SQLSERVER_EXT.SYS_VIEWS
SYS.ALL_VIEWS	AWS_SQLSERVER_EXT.SYS_ALL_VIEWS
SYS.TYPES	AWS_SQLSERVER_EXT.SYS_TYPES
SYS.COLUMNS	AWS_SQLSERVER_EXT.SYS_COLUMNS
SYS.ALL_COLUMNS	AWS_SQLSERVER_EXT.SYS_ALL_COLUMNS
SYS.FOREIGN_KEYS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEYS
SYS.SYSFOREIGNKEYS	AWS_SQLSERVER_EXT.SYS_SYSFOREIGNKEYS
SYS.FOREIGN_KEY_COLUMNS	AWS_SQLSERVER_EXT.SYS_FOREIGN_KEY_COLUMNS
SYS.KEY_CONSTRAINTS	AWS_SQLSERVER_EXT.SYS_KEY_CONSTRAINTS
SYS.IDENTITY_COLUMNS	AWS_SQLSERVER_EXT.SYS_IDENTITY_COLUMNS
SYS.PROCEDURES	AWS_SQLSERVER_EXT.SYS_PROCEDURES
SYS.INDEXES	AWS_SQLSERVER_EXT.SYS_INDEXES

Casos de uso de MS SQL Server	Sustitución de PostgreSQL
SYS.SYSINDEXES	AWS_SQLSERVER_EXT.SYS_SYSINDEXES
SYS.OBJECTS	AWS_SQLSERVER_EXT.SYS_OBJECTS
SYS.ALL_OBJECTS	AWS_SQLSERVER_EXT.SYS_ALL_OBJECTS
SYS.SYSOBJECTS	AWS_SQLSERVER_EXT.SYS_SYSOBJECTS
SYS.SQL_MODULES	AWS_SQLSERVER_EXT.SYS_SQL_MODULES
SYS.DATABASES	AWS_SQLSERVER_EXT.SYS_DATABASES
INFORMATION_SCHEMA.SCHEMATA	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_SCHEMATA
INFORMATION_SCHEMA.VIEWS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_VIEWS
INFORMATION_SCHEMA.TABLES	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLES
INFORMATION_SCHEMA.COLUMNS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_COLUMNS
INFORMATION_SCHEMA.CHECK_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_CHECK_CONSTRAINTS
INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_REFERENTIAL_CONSTRAINTS
INFORMATION_SCHEMA.TABLE_CONSTRAINTS	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_TABLE_CONSTRAINTS
INFORMATION_SCHEMA.KEY_COLUMN_USAGE	AWS_SQLSERVER_EXT.INFORMATION_SCHEMA_KEY_COLUMN_USAGE

Casos de uso de MS SQL Server	Sustitución de PostgreSQL
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

Usar un paquete de extensión de AWS SCT para simular el Agente SQL Server en PostgreSQL

El Agente SQL Server es un servicio de Microsoft Windows que ejecuta trabajos de SQL Server. El Agente SQL Server ejecuta trabajos según una programación, en respuesta a un evento concreto o bajo demanda. Para obtener más información sobre el Agente SQL Server, consulte Agente SQL Server en la documentación de Microsoft.

PostgreSQL no incluye un equivalente para el Agente SQL Server. Para simular las funciones del Agente SQL Server, AWS SCT crea un paquete de extensión. Este paquete de extensión utiliza AWS Lambda y Amazon CloudWatch. AWS Lambda implementa la interfaz que se utiliza para administrar las programaciones y ejecutar los trabajos. Amazon CloudWatch mantiene las reglas de programación.

AWS Lambda y Amazon CloudWatch utilizan un parámetro JSON para interactuar. Este parámetro JSON presenta la siguiente estructura.

```
{
    "mode": mode,
    "parameters": {
        list of parameters
```

```
    },
    "callback": procedure name
}
```

En el ejemplo anterior, *mode* es el tipo de tarea y *list of parameters* es un conjunto de parámetros que dependen del tipo de tarea. Además, *procedure name* es el nombre del procedimiento que se ejecuta una vez finalizada la tarea.

AWS SCT usa una función de Lambda para controlar y ejecutar los trabajos. La regla de CloudWatch inicia la ejecución del trabajo y proporciona la información necesaria para iniciarlo. Cuando se activa la regla de CloudWatch, esta inicia la función de Lambda con los parámetros de la regla.

Para crear un trabajo sencillo que llame a un procedimiento, utilice el siguiente formato.

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

Para crear un trabajo con varios pasos, utilice el siguiente formato.

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

Para simular el comportamiento del Agente SQL Server en PostgreSQL, el paquete de extensión de AWS SCT también crea las siguientes tablas y procedimientos.

Tablas que simulan el Agente SQL Server en PostgreSQL

Para simular el Agente SQL Server, el paquete de extensión utiliza las siguientes tablas:

sysjobs

Almacena la información sobre los trabajos.

sysjobsteps

Almacena la información sobre los pasos de un trabajo.

sysschedules

Almacena la información sobre las programaciones de los trabajos.

sysjobschedules

Almacena la información de programación de los trabajos individuales.

sysjobhistory

Almacena la información sobre las ejecuciones de los trabajos programados.

Procedimientos que simulan el Agente SQL Server en PostgreSQL

Para simular el Agente SQL Server, el paquete de extensión utiliza los siguientes procedimientos:

sp_add_job

Agrega un trabajo nuevo.

sp_add_jobstep

Agrega un paso a un trabajo.

sp_add_schedule

Crea una regla de programación nueva en Amazon CloudWatch. Puede usar esta programación con cualquier número de trabajos.

sp_attach_schedule

Establece una programación para el trabajo seleccionado.

sp_add_jobschedule

Crea una regla de programación para un trabajo en Amazon CloudWatch y establece el destino de esta regla.

sp_update_job

Actualiza los atributos del trabajo creado anteriormente.

sp_update_jobstep

Actualiza los atributos del paso de un trabajo.

sp_update_schedule

Actualiza los atributos de una regla de programación en Amazon CloudWatch.

sp_update_jobschedule

Actualiza los atributos de la programación del trabajo especificado.

sp_delete_job

Elimina un trabajo

sp_delete_jobstep

Elimina un paso de un trabajo.

sp_delete_schedule

Elimina una programación.

sp_delete_jobschedule

Elimina la regla de programación del trabajo especificado de Amazon CloudWatch.

sp_detach_schedule

Elimina la asociación entre una programación y un trabajo.

get_jobs, update_job

Procedimientos internos que interactúan con AWS Elastic Beanstalk.

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

Procedimientos internos que comprueban la configuración.

Sintaxis para procedimientos que simulan el Agente SQL Server en PostgreSQL

El procedimiento `aws_sqlserver_ext.sp_add_job` del paquete de extensión simula el procedimiento `msdb.dbo.sp_add_job`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_add_jobstep` del paquete de extensión simula el procedimiento `msdb.dbo.sp_add_jobstep`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_add_schedule` del paquete de extensión simula el procedimiento `msdb.dbo.sp_add_schedule`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_attach_schedule` del paquete de extensión simula el procedimiento `msdb.dbo.sp_attach_schedule`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_add_jobschedule` del paquete de extensión simula el procedimiento `msdb.dbo.sp_add_jobschedule`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_delete_job` del paquete de extensión simula el procedimiento `msdb.dbo.sp_delete_job`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_delete_jobstep` del paquete de extensión simula el procedimiento `msdb.dbo.sp_delete_jobstep`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de Microsoft.

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

El procedimiento `aws_sqlserver_ext.sp_delete_jobschedule` del paquete de extensión simula el procedimiento `msdb.dbo.sp_delete_jobschedule`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_delete_schedule` del paquete de extensión simula el procedimiento `msdb.dbo.sp_delete_schedule`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_detach_schedule` del paquete de extensión simula el procedimiento `msdb.dbo.sp_detach_schedule`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_update_job` del paquete de extensión simula el procedimiento `msdb.dbo.sp_update_job`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_update_jobschedule` del paquete de extensión simula el procedimiento `msdb.dbo.sp_update_jobschedule`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_update_jobstep` del paquete de extensión simula el procedimiento `msdb.dbo.sp_update_jobstep`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sp_update_schedule` del paquete de extensión simula el procedimiento `msdb.dbo.sp_update_schedule`. Para obtener más información sobre el Agente SQL Server de origen, consulte Documentación técnica de 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
```

Ejemplos de uso de procedimientos que simulan el Agente SQL Server en PostgreSQL

Para agregar un trabajo nuevo, utilice el procedimiento `aws_sqlserver_ext.sp_add_job` que se muestra a continuación.

```
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');
```

Para agregar un paso de trabajo nuevo, utilice el procedimiento `aws_sqlserver_ext.sp_add_jobstep` que se muestra a continuación.

```
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');
```

Para agregar una programación simple, utilice el procedimiento `aws_sqlserver_ext.sp_add_schedule` que se muestra a continuación.

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

Para establecer una programación para un trabajo, utilice el procedimiento `aws_sqlserver_ext.sp_attach_schedule` que se muestra a continuación.

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

Para agregar una programación para un trabajo, utilice el procedimiento `aws_sqlserver_ext.sp_add_jobschedule` que se muestra a continuación.

```
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);
```

Ejemplos de casos de uso para simular el Agente SQL Server en PostgreSQL

Si el código de la base de datos de origen utiliza el Agente SQL Server para ejecutar trabajos, puede utilizar el paquete de extensión de SQL Server a PostgreSQL para que AWS SCT convierta este código a PostgreSQL. El paquete de extensión utiliza funciones AWS Lambda para simular el comportamiento del Agente SQL Server.

Puede crear una función AWS Lambda nueva o registrar una función existente.

Para crear una función AWS Lambda nueva

1. En AWS SCT, en el árbol de la base de datos de destino, abra el menú contextual (clic secundario) y seleccione Aplicar paquete de extensión para y, a continuación, elija PostgreSQL.

   Aparecerá el asistente de paquete de extensión.
2. En la pestaña Servicio de simulación del Agente SQL Server, haga lo siguiente:

   - Seleccione Crear una función AWS Lambda.
   - En Iniciar sesión en la base de datos, introduzca el nombre del usuario de la base de datos de destino.
   - En Contraseña de la base de datos, introduzca la contraseña del nombre de usuario que introdujo en el paso anterior.
   - En carpeta de la biblioteca Python, introduzca la ruta a la carpeta de la biblioteca de Python.
   - Seleccione Crear función AWS Lambda y, a continuación, elija Siguiente.

Para registrar una función AWS Lambda que implementó anteriormente

- Ejecute el siguiente script en la base de datos de destino.

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

  En el ejemplo anterior, *ARN* es el nombre de recurso de Amazon (ARN) de la función AWS Lambda implementada.

En el siguiente ejemplo se crea una tarea simple que consta de un paso. Cada cinco minutos, esta tarea ejecuta la función `job_example` creada anteriormente. Esta función inserta los registros en la tabla `job_example_table`.

Para crear esta tarea simple

1. Cree un trabajo mediante la función `aws_sqlserver_ext.sp_add_job` según se muestra a continuación.

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

2. Cree un paso de trabajo mediante la función `aws_sqlserver_ext.sp_add_jobstep` según se muestra a continuación.

```
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;');
```

   El paso de trabajo especifica lo que hace la función.

3. Cree una programación para el trabajo mediante la función `aws_sqlserver_ext.sp_add_jobschedule` según se muestra a continuación.

```
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 */);
```

   El paso de trabajo especifica lo que hace la función.

Para eliminar este trabajo, utilice la función `aws_sqlserver_ext.sp_delete_job` según se muestra a continuación.

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

Usar un paquete de extensión de AWS SCT para simular el correo electrónico de base de datos de SQL Server en PostgreSQL

Puede utilizar el correo electrónico de base de datos de SQL Server para enviar mensajes de correo electrónico a los usuarios desde el motor de base de datos de SQL Server o la instancia administrada de Azure SQL. Estos mensajes de correo electrónico pueden contener resultados de consultas o incluir archivos de cualquier recurso de la red. Para obtener más información sobre el correo electrónico de base de datos de SQL Server, consulte Documentación técnica de Microsoft.

PostgreSQL no incluye un equivalente para el correo electrónico de base de datos de SQL Server. Para simular las funciones del correo electrónico de base de datos de SQL Server, AWS SCT crea un paquete de extensión. Este paquete de extensión utiliza AWS Lambda y Amazon Simple Email Service (Amazon SES). AWS Lambda proporciona a los usuarios una interfaz para interactuar con el servicio de envío de correo electrónico de Amazon SES. Para configurar esta interacción, agregue el nombre de recurso de Amazon (ARN) de su función de Lambda.

En una nueva cuenta de correo electrónico, utilice el siguiente comando.

```
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;
```

Para agregar el ARN de la función de Lambda a la cuenta de correo electrónico existente, utilice el siguiente comando.

```
do
```

```
$$
begin
PERFORM sysmail_update_account_sp (
    par_account_name :='existind_account_name',
    par_mailserver_type := 'AWSLAMBDA'
    par_mailserver_name := 'ARN'
);
end;
$$ language plpgsql;
```

En los ejemplos anteriores, *ARN* es el ARN de la función de Lambda.

Para simular el comportamiento del correo electrónico de base de datos de SQL Server en PostgreSQL, el paquete de extensión de AWS SCT también crea las siguientes tablas y procedimientos.

Tablas que simulan el correo electrónico de base de datos de SQL Server en PostgreSQL

Para simular el correo electrónico de base de datos de SQL Server, el paquete de extensión utiliza las siguientes tablas:

sysmail_account

Almacena la información sobre las cuentas de correo electrónico.

sysmail_profile

Almacena la información sobre los perfiles de usuario.

sysmail_server

Almacena la información sobre los servidores de correo electrónico.

sysmail_mail_items

Almacena la lista de los mensajes de correo electrónico.

sysmail_attachments

Contiene una fila para cada archivos adjunto de correo electrónico.

sysmail_log

Almacena la información de servicio sobre el envío de mensajes de correo electrónico.

sysmail_profileaccount

Almacena la información sobre los perfiles de usuario y las cuentas de correo electrónico.

Vistas que simulan el correo electrónico de base de datos de SQL Server en PostgreSQL

Para simular el correo electrónico de base de datos de SQL Server, AWS SCT crea las siguientes vistas en la base de datos de PostgreSQL para garantizar la compatibilidad. El paquete de extensión no las usa, pero el código convertido puede consultar estas vistas.

sysmail_allitems

Incluye una lista de todos los correos electrónicos.

sysmail_faileditems

Incluye una lista de los correos electrónicos que no se pudieron enviar.

sysmail_sentitems

Incluye una lista de los correos electrónicos enviados.

sysmail_unsentitems

Incluye una lista de los correos electrónicos que aún no se han enviado.

sysmail_mailattachments

Incluye una lista de archivos adjuntos.

Procedimientos que simulan el correo electrónico de base de datos de SQL Server en PostgreSQL

Para simular el correo electrónico de base de datos de SQL Server, el paquete de extensión utiliza los siguientes procedimientos:

sp_send_dbmail

Envía un correo electrónico a los destinatarios especificados.

sysmail_add_profile_sp

Crea un perfil de usuario nuevo.

sysmail_add_account_sp

Crea una cuenta de correo electrónico nueva que almacena información como las credenciales del protocolo simple de transferencia de correo (SMTP), etc.

sysmail_add_profileaccount_sp

Agrega una cuenta de correo electrónico al perfil de usuario especificado.

sysmail_update_profile_sp

Cambia los atributos del perfil de usuario, como la descripción, el nombre, etc.

sysmail_update_account_sp

Cambia la información de la cuenta de correo electrónico existente.

sysmail_update_profileaccount_sp

Actualiza la información de la cuenta de correo electrónico en el perfil de usuario especificado.

sysmail_delete_profileaccount_sp

Elimina una cuenta de correo electrónico del perfil de usuario especificado.

sysmail_delete_account_sp

Elimina la cuenta de correo electrónico.

sysmail_delete_profile_sp

Elimina el perfil de usuario.

sysmail_delete_mailitems_sp

Elimina los correos electrónicos de las tablas internas.

sysmail_help_profile_sp

Muestra información sobre el perfil del usuario.

sysmail_help_account_sp

Muestra información sobre la cuenta de correo electrónico.

sysmail_help_profileaccount_sp

Muestra información sobre las cuentas de correo electrónico asociadas al perfil de usuario.

sysmail_dbmail_json

Procedimiento interno que genera solicitudes JSON para funciones AWS Lambda.

sysmail_verify_profile_sp, sysmail_verify_account_sp, sysmail_verify_addressparams_sp

Procedimientos internos que comprueban la configuración.

sp_get_dbmail, sp_set_dbmail, sysmail_dbmail_xml

Procedimientos internos obsoletos.

Sintaxis para procedimientos que simulan el correo electrónico de base de datos de SQL Server en PostgreSQL

El procedimiento `aws_sqlserver_ext.sp_send_dbmail` del paquete de extensión simula el procedimiento `msdb.dbo.sp_send_dbmail`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sysmail_delete_mailitems_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_delete_mailitems_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de Microsoft.

```
par_sent_before timestamp = NULL::timestamp without time zone,
par_sent_status varchar = NULL::character varying,
```

```
out returncode integer
```

El procedimiento `aws_sqlserver_ext.sysmail_add_profile_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_add_profile_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de Microsoft.

```
par_profile_name varchar,
par_description varchar = NULL::character varying,
out par_profile_id integer,
out returncode integer
```

El procedimiento `aws_sqlserver_ext.sysmail_add_account_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_add_account_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sysmail_add_profileaccount_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_add_profileaccount_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sysmail_help_profile_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_help_profile_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de Microsoft.

```
par_profile_id integer = NULL::integer,
par_profile_name varchar = NULL::character varying,
out returncode integer
```

El procedimiento `aws_sqlserver_ext.sysmail_update_profile_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_update_profile_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de Microsoft.

```
par_profile_id integer = NULL::integer,
par_profile_name varchar = NULL::character varying,
par_description varchar = NULL::character varying,
out returncode integer
```

El procedimiento `aws_sqlserver_ext.sysmail_delete_profile_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_delete_profile_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de Microsoft.

```
par_profile_id integer = NULL::integer,
par_profile_name varchar = NULL::character varying,
par_force_delete smallint = 1,
out returncode integer
```

El procedimiento `aws_sqlserver_ext.sysmail_help_account_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_help_account_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de Microsoft.

```
par_account_id integer = NULL::integer,
```

```
par_account_name varchar = NULL::character varying,
out returncode integer
```

El procedimiento `aws_sqlserver_ext.sysmail_update_account_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_update_account_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sysmail_delete_account_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_delete_account_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de Microsoft.

```
par_account_id integer = NULL::integer,
par_account_name varchar = NULL::character varying,
out returncode integer
```

El procedimiento `aws_sqlserver_ext.sysmail_help_profileaccount_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_help_profileaccount_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sysmail_update_profileaccount_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_update_profileaccount_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de 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
```

El procedimiento `aws_sqlserver_ext.sysmail_delete_profileaccount_sp` del paquete de extensión simula el procedimiento `msdb.dbo.sysmail_delete_profileaccount_sp`. Para obtener más información sobre el correo electrónico de base de datos de SQL Server de origen, consulte Documentación técnica de 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
```

Ejemplos de uso de procedimientos que simulan el correo electrónico de base de datos de SQL Server en PostgreSQL

Para enviar un correo electrónico, utilice el procedimiento `aws_sqlserver_ext.sp_send_dbmail` que se muestra a continuación.

```
PERFORM sp_send_dbmail (
    par_profile_name := 'Administrator',
    par_recipients := 'hello@rusgl.info',
    par_subject := 'Automated Success Message',
    par_body := 'The stored procedure finished'
);
```

El siguiente ejemplo muestra cómo enviar un correo electrónico con resultados de consulta.

```
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'
);
```

En el siguiente ejemplo de código se muestra cómo enviar un correo electrónico con código HTML.

```
DECLARE var_tableHTML TEXT;
SET var_tableHTML := CONCAT(
    '<H1>Work Order Report</H1>',
    '<table border="1">',
    '<tr><th>Work Order ID</th><th>Product ID</th>',
    '<th>Name</th><th>Order Qty</th><th>Due Date</th>',
    '<th>Expected Revenue</th></tr>',
    '</table>'
);
PERFORM sp_send_dbmail (
    par_recipients := 'hello@rusgl.info',
    par_subject := 'Work Order List',
    par_body := var_tableHTML,
    par_body_format := 'HTML'
);
```

Para eliminar correos electrónicos, utilice el procedimiento `aws_sqlserver_ext.sysmail_delete_mailitems_sp` que se muestra a continuación.

```
DECLARE var_GETDATE datetime;
SET var_GETDATE = NOW();
PERFORM sysmail_delete_mailitems_sp (
    par_sent_before := var_GETDATE
);
```

En el siguiente ejemplo se muestra cómo se eliminan los correos electrónicos más antiguos.

```
PERFORM sysmail_delete_mailitems_sp (
    par_sent_before := '31.12.2015'
);
```

En el siguiente ejemplo se muestra cómo se eliminan todos los correos electrónicos que no se pueden enviar.

```
PERFORM sysmail_delete_mailitems_sp (
    par_sent_status := 'failed'
);
```

Para crear un perfil de usuario nuevo, utilice el procedimiento `aws_sqlserver_ext.sysmail_add_profile_sp` que se muestra a continuación.

```
PERFORM sysmail_add_profile_sp (
    profile_name := 'Administrator',
    par_description := 'administrative mail'
);
```

En el siguiente ejemplo se muestra cómo crear un perfil nuevo y guardar el identificador de perfil único en una variable.

```
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;
```

Para crear una cuenta de correo electrónico nueva, utilice el procedimiento `aws_sqlserver_ext.sysmail_add_account_sp` como se muestra a continuación.

```
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'
);
```

Para agregar una cuenta de correo electrónico al perfil de usuario, utilice el procedimiento `aws_sqlserver_ext.sysmail_add_profileaccount_sp` según se muestra a continuación.

```
PERFORM sysmail_add_profileaccount_sp (
    par_account_name := 'Administrator',
    par_account_name := 'Audit Account',
    par_sequence_number := 1
);
```

Ejemplos de casos de uso para simular el correo electrónico de base de datos de SQL Server en PostgreSQL

Si el código de la base de datos de origen utiliza el correo electrónico de base de datos de SQL Server para enviar correos electrónicos, puede utilizar el paquete de extensión de AWS SCT para convertir este código a PostgreSQL.

Para enviar un correo electrónico desde su base de datos de PostgreSQL

1. Cree y configure su función AWS Lambda.
2. Aplique el paquete de extensión de AWS SCT.
3. Cree un perfil de usuario mediante la función `sysmail_add_profile_sp` según se muestra a continuación.
4. Cree una cuenta de correo electrónico mediante la función `sysmail_add_account_sp` según se muestra a continuación.
5. Agregue esta cuenta de correo electrónico a su perfil de usuario mediante la función `sysmail_add_profileaccount_sp` según se muestra a continuación.

```
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. Envíe un correo electrónico mediante la función `sp_send_dbmail` según se muestra a continuación.

```
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;
```

Para ver la información sobre todos los perfiles de usuario, utilice el procedimiento `sysmail_help_profile_sp` que se muestra a continuación.

```
SELECT FROM aws_sqlserver_ext.sysmail_help_profile_sp();
```

En el siguiente ejemplo se muestra información acerca del perfil de usuario específico.

```
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');
```

Para ver la información sobre todos las cuentas de correo electrónico, utilice el procedimiento `sysmail_help_account_sp` que se muestra a continuación.

```
select from aws_sqlserver_ext.sysmail_help_account_sp();
```

En el siguiente ejemplo se muestra información acerca de la cuenta de correo electrónico específica.

```
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');
```

Para ver la información sobre todas las cuentas de correo electrónico que están asociadas a los perfiles de usuario, utilice el procedimiento `sysmail_help_profileaccount_sp` según se muestra a continuación.

```
select from aws_sqlserver_ext.sysmail_help_profileaccount_sp();
```

El siguiente ejemplo filtra los registros por identificador, nombre de perfil o nombre de cuenta.

```
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');
```

Para cambiar el nombre o la descripción del perfil de usuario, utilice el procedimiento `sysmail_update_profile_sp` según se muestra a continuación.

```
select aws_sqlserver_ext.sysmail_update_profile_sp(
    par_profile_id := 2,
    par_profile_name := 'New profile name'
);
```

Para cambiar la configuración de la cuenta de correo electrónico, utilice el procedimiento `ysmail_update_account_sp` según se muestra a continuación.

```
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'
);
```

Convertir SQL Server a Amazon RDS para SQL Server

Hay algunos aspectos que deben tenerse en cuenta al migrar un esquema y un código de SQL Server a Amazon RDS para SQL Server:

- AWS SCT puede convertir el Agente SQL Server para que proporcione programaciones, alertas y trabajos en una instancia de base de datos de Amazon RDS para SQL Server. Después de la conversión, puede utilizar una instancia de base de datos de Amazon RDS para SQL Server con SQL Server Reporting Services (SSRS), SQL Server Analysis Services (SSAS) y SQL Server Integration Services (SSIS).
- Actualmente, Amazon RDS no admite SQL Server Service Broker ni puntos de enlace de T-SQL adicionales que requieran ejecutar el comando CREATE ENDPOINT.
- Amazon RDS tiene una compatibilidad limitada con los servidores enlazados. Al convertir código de aplicación de SQL Server que utiliza servidores enlazados, AWS SCT convierte el código de aplicación. Sin embargo, asegúrese de revisar el comportamiento de los objetos que utilizan servidores enlazados antes de ejecutar el código convertido.
- Se usa siempre encendido.
- El informe de evaluación de AWS SCT proporciona métricas de servidor para la conversión. Estas métricas acerca de la instancia de SQL Server incluyen:
  - Uso de la replicación de datos.
  - Configuración de Trasvase de registros de SQL Server.
  - Uso de un clúster de conmutación por error.
  - Configuración de Correo electrónico de base de datos.
  - Uso de Servicio de búsqueda de texto completo. La búsqueda de texto completo en Amazon RDS para SQL Server es limitada y no admite la búsqueda semántica.
  - Instalación de Data Quality Services (DQS). Amazon RDS no admite DQS. Por consiguiente, recomendamos instalar SQL Server en una instancia Amazon EC2.

Privilegios para RDS para SQL Server como destino

Para migrar a RDS para SQL Server, cree un usuario de base de datos y, a continuación, otorgue los privilegios necesarios para cada base de datos. Puede utilizar el siguiente código de ejemplo.

```
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;
```

En el ejemplo anterior, sustituya *user_name* por el nombre del usuario. A continuación, sustituya *db_name* por el nombre de su base de datos de destino. Por último, sustituya *your_password* por una contraseña segura.

Orígenes de almacenamiento de datos para AWS Schema Conversion Tool

AWS SCT puede convertir esquemas de los siguientes almacenamientos de datos de origen a un destino compatible. Para obtener información acerca de los permisos, las conexiones y lo que AWS SCT puede convertir para usar con la base de datos o el almacenamiento de datos de destino, consulte los detalles de los temas que se muestran a continuación.

Temas

- Usar Amazon Redshift como origen para AWS SCT
- Utilizar Azure Synapse Analytics como origen para AWS SCT

- Usar BigQuery como origen para AWS SCT
- Utilizar la base de datos de Greenplum como origen para AWS SCT
- Utilizar Netezza como origen para AWS SCT
- Utilizar Oracle Data Warehouse como origen para AWS SCT
- Utilizar Snowflake como origen para AWS SCT
- Utilizar Microsoft SQL Server Data Warehouse como origen para AWS SCT
- Utilizar Teradata como origen para AWS SCT
- Utilizar Vertica como origen para AWS SCT

Usar Amazon Redshift como origen para AWS SCT

Puede utilizar AWS SCT para optimizar su clúster de Amazon Redshift. AWS SCT le proporciona recomendaciones sobre la selección de claves de distribución y clasificación para su clúster de Amazon Redshift. Puede considerar el proyecto de optimización de Amazon Redshift como un proyecto de AWS SCT en el que el origen y el destino apuntan a los distintos clústeres de Amazon Redshift.

Privilegios para Amazon Redshift como base de datos de origen

Los privilegios necesarios para usar Amazon Redshift como origen se enumeran a continuación:

- USAGE ON SCHEMA *<schema_name>*
- SELECT ON ALL TABLES IN SCHEMA *<schema_name>*
- SELECT ON PG_CATALOG.PG_STATISTIC
- SELECT ON SVV_TABLE_INFO
- SELECT ON TABLE STV_BLOCKLIST
- SELECT ON TABLE STV_TBL_PERM
- SELECT ON SYS_SERVERLESS_USAGE
- SELECT ON PG_DATABASE_INFO
- SELECT ON PG_STATISTIC

En los ejemplos anteriores, sustituya el marcador de posición *<schema_name>* por el nombre de la base de datos de origen.

Para conocer los privilegios necesarios para Amazon Redshift como destino, consulte Permisos para Amazon Redshift como destino.

Conectar a Amazon Redshift como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de origen de Amazon Redshift con AWS Schema Conversion Tool.

Para conectarse a una base de datos de Amazon Redshift

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Seleccione Amazon Redshift y, a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:
     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir la información de conexión de la base de datos de Amazon Redshift de origen, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.

Parámetro	Acción
Database	Introduzca el nombre de la base de datos de Amazon Redshift.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Usar SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a la base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Verify Server Certificate: seleccione esta opción para verificar el certificado del servidor mediante un almacén de confianza. • Trust store: la ubicación de un almacén de confianza que contenga certificados. Para que esta ubicación aparezca en aquí, añádala en Configuración global. Para obtener más información sobre la compatibilidad con SSL de Amazon Redshift, consulte Configurar las opciones de seguridad para las conexiones.
Store password	AWS SCT crea un almacén seguro para almacenar certificados SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.

Parámetro	Acción
Redshift driver path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Configuración de optimización de Amazon Redshift

Para editar la configuración de optimización de Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Amazon Redshift y, a continuación, Amazon Redshift — Amazon Redshift. AWS SCTmuestra todos los ajustes disponibles para la optimización de Amazon Redshift.

La configuración de optimización de Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.

- Establecer el número máximo de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift de destino.

  En El número máximo de tablas para el clúster de Amazon Redshift de destino, elija el número de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift.

  Amazon Redshift tiene cuotas que limitan las tablas de uso para los distintos tipos de nodos de clúster. Si elige Auto, AWS SCT determina el número de tablas que se van a aplicar al clúster de Amazon Redshift de destino en función del tipo de nodo. Si lo desea, elija el valor manualmente. Para obtener más información, consulte Cuotas y límites de Amazon Redshift en la Guía de administración de Amazon Redshift.

  AWS SCT convierte todas las tablas de origen, incluso si el número de tablas es superior al que el clúster de Amazon Redshift puede almacenar. AWS SCT almacena el código convertido en su proyecto y no lo aplica a la base de datos de destino. Si alcanza la cuota de clústeres de Amazon Redshift para las tablas al aplicar el código convertido, AWS SCT mostrará un mensaje de advertencia. Además, AWS SCT aplica tablas al clúster de Amazon Redshift de destino hasta que el número de tablas alcance el límite.
- Para elegir la estrategia de migración.

  AWS recomienda utilizar diferentes clústeres como origen y destino para su proyecto de optimización. Antes de iniciar el proceso de optimización de Amazon Redshift, debe crear una copia del clúster de Amazon Redshift de origen. Puede incluir los datos de origen en esta copia o crear un clúster vacío.

  En Estrategia de migración, elija Migración a una copia para incluir los datos del clúster de origen en el clúster de destino.

  En Estrategia de migración, seleccione Migración desde cero para revisar las sugerencias de optimización. Tras aceptar estas sugerencias, migre los datos de origen al clúster de destino.
- Aplicar compresión a las columnas de la tabla de Amazon Redshift. Para ello, seleccione Usar codificación de compresión.

  AWS SCT asigna automáticamente la codificación de compresión a las columnas mediante el algoritmo Amazon Redshift predeterminado. Para obtener más información, consulte Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.

De forma predeterminada, Amazon Redshift no aplica compresión a las columnas definidas como claves de clasificación y distribución. Puede cambiar este comportamiento y aplicar compresión a estas columnas. Para ello, seleccione Usar codificación de compresión para columnas KEY. Puede seleccionar esta opción solo si ha seleccionado la opción Usar codificación de compresión.

- Trabajar con optimización automática de tablas

  La optimización automática de tablas es una capacidad de autoajuste presente en Amazon Redshift que optimiza automáticamente el diseño de tablas. Para obtener más información, consulte Trabajar con optimización automática de tablas en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  Para utilizarla únicamente en la optimización automática de tablas, elija Estrategias de optimización en el panel izquierdo. A continuación, seleccione Utilizar el ajuste automático de tablas de Amazon Redshift y elija Ninguna en Estrategia inicial de selección de claves.

- Elegir las claves de clasificación y distribución según su estrategia.

  Puede elegir las claves de clasificación y distribución utilizando metadatos de Amazon Redshift, información estadística o ambas opciones. En estrategia inicial de selección de claves, en la pestaña Estrategias de optimización, elija una de las siguientes opciones:

  - Utilizar metadatos e ignorar la información estadística
  - Ignorar los metadatos, utilizar información estadística
  - Utilizar metadatos e información estadística

  En función de la opción que elija, puede seleccionar estrategias de optimización. A continuación, introduzca el valor (0—100) para cada estrategia. Estos valores definen la ponderación de cada estrategia. Con estos valores de ponderación, AWS SCT define cómo influye cada regla en la elección de las claves de distribución y clasificación. Los valores predeterminados se basan en las prácticas recomendadas de migración de AWS.

  Puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para definirla como una tabla pequeña. AWS SCT aplica el estilo de distribución ALL a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

- Configurar los detalles de la estrategia.

Además de definir la ponderación de cada estrategia de optimización, puede configurar los ajustes de optimización. Para ello, elija Optimización de la conversión.

- En Límite de columnas de clave de clasificación, introduzca el número máximo de columnas en la clave de clasificación.
- En Valor de umbral descompensado, introduzca el porcentaje (0—100) de un valor descompensado para una columna. AWS SCT excluye de la lista de candidatos a la clave de distribución las columnas con un valor de sesgo superior al umbral. AWS SCT define el valor descompensado de una columna como la relación porcentual entre el número de apariciones del valor más común y el número total de registros.
- En Consultas N principales de la tabla del historial de consultas, introduzca el número (1–100) de las consultas que se van a analizar con más frecuencia.
- En Seleccionar usuario de estadísticas, elija el usuario de la base de datos cuyas estadísticas de consulta desee analizar.

Utilizar Azure Synapse Analytics como origen para AWS SCT

Puede utilizar AWS SCT para convertir esquemas, objetos de código y código de aplicación de Azure Synapse Analytics a Amazon Redshift.

Privilegios para Azure Synapse Analytics como base de datos de origen

Se requieren los siguientes privilegios para usar un almacenamiento de datos de Azure Synapse Analytics como origen:

- VIEW DEFINITION
- VIEW DATABASE STATE

Aplicar los privilegios para cada base de datos cuyo esquema vaya a convertir.

Conectar a Azure Synapse Analytics como origen

Siga este procedimiento para conectarse a su almacenamiento de datos de Azure Synapse Analytics con AWS Schema Conversion Tool.

Para conectarse a un almacenamiento de datos de Azure Synapse Analytics como origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.

2. Elija Azure Synapse Analytics y, a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.

3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.

4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.

   - Para introducir manualmente la información de conexión del almacenamiento de datos de Azure Synapse Analytics, siga estas instrucciones:

Parámetro	Acción
Server name	Escriba el nombre del servicio de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
SQL pool	Introduzca el nombre del grupo de Azure SQL.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña

Parámetro	Acción
	para conectarse a su base de datos de origen, según sea necesario.
Usar SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a la base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Trust server certificate: seleccione esta opción para confiar en el certificado del servidor. • Trust store: un almacén de confianza que haya configura do en Configuración global.
Store password	AWS SCT crea un almacén seguro para almacenar certifica dos SSL y contraseñas de la base de datos. Al activar esta opción, puede almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Configuración de conversión de Azure Synapse Analytics a Amazon Redshift

Para editar la configuración de conversión de Azure Synapse Analytics a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Azure Synapse y, a continuación, Azure Synapse — Amazon Redshift. AWS SCTmuestra todos los ajustes disponibles para la conversión de Azure Synapse Analytics a Amazon Redshift.

La configuración de conversión de Azure Synapse Analytics a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.

- Establecer el número máximo de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift de destino.

  En El número máximo de tablas para el clúster de Amazon Redshift de destino, elija el número de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift.

  Amazon Redshift tiene cuotas que limitan las tablas de uso para los distintos tipos de nodos de clúster. Si elige Auto, AWS SCT determina el número de tablas que se van a aplicar al clúster de Amazon Redshift de destino en función del tipo de nodo. Si lo desea, elija el valor manualmente. Para obtener más información, consulte Cuotas y límites de Amazon Redshift en la Guía de administración de Amazon Redshift.

  AWS SCT convierte todas las tablas de origen, aunque sean más de las que su clúster de Amazon Redshift puede almacenar. AWS SCTalmacena el código convertido en su proyecto y no lo aplica a la base de datos de destino. Si alcanza la cuota del clúster de Amazon Redshift para las tablas al aplicar el código convertido, AWS SCT mostrará un mensaje de advertencia. Además, AWS SCT aplica tablas a su clúster de Amazon Redshift de destino hasta que el número de tablas alcanza el límite.
- Migrar particiones de la tabla de origen a tablas independientes en Amazon Redshift. Para ello, seleccione Usar la vista UNION ALL e introduzca el número máximo de tablas de destino que AWS SCT puede crear para una sola tabla de origen.

  Amazon Redshift no admite particionamiento de tablas. Para simular este comportamiento y agilizar la ejecución de las consultas, AWS SCT puede migrar cada partición de la tabla de origen a una tabla independiente en Amazon Redshift. A continuación, AWS SCT crea una vista que incluye los datos de todas estas tablas.

  AWS SCT determina automáticamente el número de particiones de la tabla de origen. Según el tipo de particionamiento de la tabla de origen, este número puede superar la cuota de tablas que

puede aplicar a su clúster de Amazon Redshift. Para evitar alcanzar esta cuota, introduzca el número máximo de tablas de destino que AWS SCT puede crear para las particiones de una sola tabla de origen. La opción predeterminada es 368 tablas, lo que representa una partición para 366 días de un año y dos tablas para las particiones `NO RANGE` y `UNKNOWN`.

- Aplicar compresión a las columnas de la tabla de Amazon Redshift. Para ello, seleccione Usar codificación de compresión.

  AWS SCT asigna automáticamente la codificación de compresión a las columnas mediante el algoritmo Amazon Redshift predeterminado. Para obtener más información, consulte Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  De forma predeterminada, Amazon Redshift no aplica compresión a las columnas definidas como claves de clasificación y distribución. Puede cambiar este comportamiento y aplicar compresión a estas columnas. Para ello, seleccione Usar codificación de compresión para columnas KEY. Puede seleccionar esta opción solo si selecciona la opción Usar codificación de compresión.

Configuración de optimización de la conversión de Azure Synapse Analytics a Amazon Redshift

Para editar la configuración de optimización de la conversión de Azure Synapse Analytics a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Azure Synapse y, a continuación, Azure Synapse — Amazon Redshift. En el panel izquierdo, elija Estrategias de optimización. AWS SCT muestra la configuración de optimización de la conversión para la conversión de Azure Synapse Analytics a Amazon Redshift.

La configuración de optimización de la conversión de Azure Synapse Analytics a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Trabajar con optimización automática de tablas. Para ello, seleccione Utilizar el ajuste automático de tablas de Amazon Redshift.

  La optimización automática de tablas es una capacidad de autoajuste presente en Amazon Redshift que optimiza automáticamente el diseño de tablas. Para obtener más información, consulte Trabajar con optimización automática de tablas en la Guía para desarrolladores de bases de datos de Amazon Redshift.

Para confiar únicamente en la optimización automática de las tablas, elija Ninguna en Estrategia inicial de selección de claves.

- Elegir las claves de clasificación y distribución según su estrategia.

  Puede elegir las claves de clasificación y distribución utilizando metadatos de Amazon Redshift, información estadística o ambas opciones. En estrategia inicial de selección de claves, en la pestaña Estrategias de optimización, elija una de las siguientes opciones:

  - Utilizar metadatos e ignorar la información estadística
  - Ignorar los metadatos, utilizar información estadística
  - Utilizar metadatos e información estadística

  En función de la opción que elija, puede seleccionar estrategias de optimización. A continuación, introduzca el valor (0—100) para cada estrategia. Estos valores definen la ponderación de cada estrategia. Con estos valores de ponderación, AWS SCT define cómo influye cada regla en la elección de las claves de distribución y clasificación. Los valores predeterminados se basan en las prácticas recomendadas de migración de AWS.

  Puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para definirla como una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

- Configurar los detalles de la estrategia.

  Además de definir la ponderación de cada estrategia de optimización, puede configurar los ajustes de optimización. Para ello, elija Optimización de la conversión.

  - En Límite de columnas de clave de clasificación, introduzca el número máximo de columnas en la clave de clasificación.
  - En Valor de umbral descompensado, introduzca el porcentaje (0—100) de un valor descompensado para una columna. AWS SCT excluye de la lista de candidatos a la clave de distribución las columnas con un valor de sesgo superior al umbral. AWS SCT define el valor descompensado de una columna como la relación porcentual entre el número de apariciones del valor más común y el número total de registros.
  - En Consultas N principales de la tabla del historial de consultas, introduzca el número (1–100) de las consultas que se van a analizar con más frecuencia.

- En Seleccionar usuario de estadísticas, elija el usuario de la base de datos cuyas estadísticas de consulta desee analizar.

Además, en la pestaña Estrategias de optimización, puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para considerarla una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

Usar BigQuery como origen para AWS SCT

Puede utilizar AWS SCT para convertir esquemas, objetos de código y código de aplicación de BigQuery a Amazon Redshift.

Privilegios para BigQuery como origen

Para usar un almacenamiento de datos de BigQuery como origen en AWS SCT, cree una cuenta de servicio. En Google Cloud, las aplicaciones usan cuentas de servicio para realizar llamadas a la API autorizadas. Las cuentas de servicio son diferentes de las cuentas de usuario. Para obtener más información, consulte Cuentas de servicio en la documentación sobre Google Cloud Identity and Access Management.

Asegúrese de asignar las siguientes funciones a su cuenta de servicio:

- `BigQuery Admin`
- `Storage Admin`

El rol `BigQuery Admin` proporciona permisos para administrar todos los recursos del proyecto. AWS SCT usa esta función para cargar tus metadatos de BigQuery en el proyecto de migración.

El rol `Storage Admin` otorga el control total de los objetos y buckets de datos. Puede encontrar este rol en.`Cloud Storage`. AWS SCT usa esta función para extraer sus datos de BigQuery y luego cargarlos en Amazon Redshift.

Para crear un archivo de clave de cuenta de servicio

1. Inicie sesión en la consola de administración de Google Cloud en https://console.cloud.google.com/.

2. En la página API de BigQuery, seleccione Habilitar. Omita este paso si ve API habilitada.
3. En la página Cuentas de servicio, seleccione su proyecto y, a continuación, seleccione Crear cuenta de servicio.
4. En la página Detalles de la cuenta de servicio, introduzca un valor descriptivo para Nombre de la cuenta de servicio. Seleccione Crear y continuar. Se abre la página Otorgar a esta cuenta de servicio acceso al proyecto.
5. En Seleccionar un rol, seleccione BigQuery y, a continuación, seleccione Administrador de BigQuery.
6. Elija Agregar otro rol. En Seleccionar un rol, seleccione Cloud Storage y, a continuación, seleccione Administrador de almacenamiento.
7. Elija Continur y, a continuación, elija Guardar.
8. En la página Cuentas de servicio, elija la cuenta de servicio que creó.
9. Seleccione Claves y, a continuación, seleccione Crear clave nueva para Agregar clave.
10. Elija JSON y, a continuación, elija Crear. Elija la carpeta para guardar su clave privada o seleccione la carpeta predeterminada para las descargas en su navegador.

Para exportar datos de un almacenamiento de datos de BigQuery, AWS SCT usa la carpeta del bucket de Google Cloud Storage. Cree este bucket antes de iniciar la migración de datos. Introduzca la ruta a la carpeta del bucket de Google Cloud Storage en el cuadro de diálogo Crear tarea local. Para obtener más información, consulte Crear, ejecutar y supervisar una AWS SCT tarea.

Conectar a BigQuery como origen

Utilice el siguiente procedimiento para conectarse a su proyecto de BigQuery de origen con AWS Schema Conversion Tool.

Para conectaree a un almacenamiento de datos de origen de BigQuery

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Seleccione BigQuery y, a continuación, seleccione Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su proyecto de BigQuery. AWS SCT muestra este nombre en el árbol del panel izquierdo.

4. En Ruta de la clave, introduzca la ruta al archivo de la clave de la cuenta de servicio. Para obtener más información sobre la creación de este archivo, consulte Privilegios para BigQuery como origen.
5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su proyecto de BigQuery de origen.
6. Elija Conectar para conectarse a su proyecto de BigQuery de origen.

Limitaciones en el uso de BigQuery como origen para AWS SCT

Al utilizar BigQuery como origen para AWS SCT se aplican las siguientes limitaciones:

- AWS SCT no admite la conversión de subconsultas en funciones analíticas.
- No puede usar AWS SCT para convertir instrucciones `SELECT AS STRUCT` y `SELECT AS VALUE` de BigQuery.
- AWS SCT no admite la conversión de los siguientes tipos de funciones:
  - Approximate aggregate
  - Bit
  - Debugging
  - Federated query
  - Geography
  - Hash
  - Mathematical
  - Net
  - Statistical aggregate
  - UUID
- AWS SCT proporciona soporte limitado para la conversión de funciones de cadena.
- AWS SCT no admite la conversión de operadores `UNNEST`.
- No puede convertir operaciones de unión correlacionadas en AWS SCT.
- AWS SCT no admite la conversión de cláusulas`QUALIFY`, `WINDOW`, `LIMIT` y `OFFSET`.
- No puede utilizar AWS SCT para convertir expresiones de tabla comunes recursivas.
- AWS SCT no admite la conversión de instrucciones `INSERT` con subconsultas dentro de cláusulas `VALUES`.

- AWS SCT no admite la conversión de instrucciones `UPDATE` para campos anidados y registros repetidos.
- No puede utilizar AWS SCT para convertir tipos de datos `STRUCT` y `ARRAY`.

Configuración de conversión de BigQuery a Amazon Redshift

Para editar la configuración de conversión de BigQuery a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija BigQuery y, a continuación, elija BigQuery — Amazon Redshift. AWS SCT muestra todos los ajustes disponibles para la conversión de BigQuery a Amazon Redshift.

La configuración de conversión de BigQuery a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.
- Establecer el número máximo de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift de destino.

  En El número máximo de tablas para el clúster de Amazon Redshift de destino, elija el número de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift.

  Amazon Redshift tiene cuotas que limitan las tablas de uso para los distintos tipos de nodos de clúster. Si elige Auto, AWS SCT determina el número de tablas que se van a aplicar al clúster de Amazon Redshift de destino en función del tipo de nodo. Si lo desea, elija el valor manualmente. Para obtener más información, consulte Cuotas y límites de Amazon Redshift en la Guía de administración de Amazon Redshift.

  AWS SCT convierte todas las tablas de origen, aunque sean más de las que su clúster de Amazon Redshift puede almacenar. AWS SCTalmacena el código convertido en su proyecto y no lo aplica

a la base de datos de destino. Si alcanza la cuota del clúster de Amazon Redshift para las tablas al aplicar el código convertido, AWS SCT mostrará un mensaje de advertencia. Además, AWS SCT aplica tablas a su clúster de Amazon Redshift de destino hasta que el número de tablas alcanza el límite.

- Aplicar compresión a las columnas de la tabla de Amazon Redshift. Para ello, seleccione Usar codificación de compresión.

  AWS SCT asigna automáticamente la codificación de compresión a las columnas mediante el algoritmo Amazon Redshift predeterminado. Para obtener más información, consulte Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  De forma predeterminada, Amazon Redshift no aplica compresión a las columnas definidas como claves de clasificación y distribución. Puede cambiar este comportamiento y aplicar compresión a estas columnas. Para ello, seleccione Usar codificación de compresión para columnas KEY. Puede seleccionar esta opción solo si selecciona la opción Usar codificación de compresión.

Configuración de optimización de la conversión de BigQuery a Amazon Redshift

Para editar la configuración de optimización de la conversión de BigQuery a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, seleccione Google BigQuery y, a continuación, Google BigQuery — Amazon Redshift. En el panel izquierdo, elija Estrategias de optimización. AWS SCT muestra la configuración de optimización de la conversión para la conversión de BigQuery a Amazon Redshift.

La configuración de optimización de la conversión de BigQuery a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Trabajar con optimización automática de tablas. Para ello, seleccione Utilizar el ajuste automático de tablas de Amazon Redshift.

  La optimización automática de tablas es una capacidad de autoajuste presente en Amazon Redshift que optimiza automáticamente el diseño de tablas. Para obtener más información, consulte Trabajar con optimización automática de tablas en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  Para confiar únicamente en la optimización automática de las tablas, elija Ninguna en Estrategia inicial de selección de claves.

- Elegir las claves de clasificación y distribución según su estrategia.

  Puede elegir las claves de clasificación y distribución utilizando metadatos de Amazon Redshift, información estadística o ambas opciones. En estrategia inicial de selección de claves, en la pestaña Estrategias de optimización, elija una de las siguientes opciones:

  - Utilizar metadatos e ignorar la información estadística
  - Ignorar los metadatos, utilizar información estadística
  - Utilizar metadatos e información estadística

  En función de la opción que elija, puede seleccionar estrategias de optimización. A continuación, introduzca el valor (0—100) para cada estrategia. Estos valores definen la ponderación de cada estrategia. Con estos valores de ponderación, AWS SCT define cómo influye cada regla en la elección de las claves de distribución y clasificación. Los valores predeterminados se basan en las prácticas recomendadas de migración de AWS.

  Puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para definirla como una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

- Configurar los detalles de la estrategia.

  Además de definir la ponderación de cada estrategia de optimización, puede configurar los ajustes de optimización. Para ello, elija Optimización de la conversión.

  - En Límite de columnas de clave de clasificación, introduzca el número máximo de columnas en la clave de clasificación.
  - En Valor de umbral descompensado, introduzca el porcentaje (0—100) de un valor descompensado para una columna. AWS SCT excluye de la lista de candidatos a la clave de distribución las columnas con un valor de sesgo superior al umbral. AWS SCT define el valor descompensado de una columna como la relación porcentual entre el número de apariciones del valor más común y el número total de registros.
  - En Consultas N principales de la tabla del historial de consultas, introduzca el número (1–100) de las consultas que se van a analizar con más frecuencia.
  - En Seleccionar usuario de estadísticas, elija el usuario de la base de datos cuyas estadísticas de consulta desee analizar.

Además, en la pestaña Estrategias de optimización, puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para considerarla una tabla pequeña. AWS SCT aplica el estilo de distribución ALL a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

Utilizar la base de datos de Greenplum como origen para AWS SCT

Puede utilizar AWS SCT para convertir esquemas, objetos de código y código de aplicación de la base de datos de Greenplum a Amazon Redshift.

Privilegios para la base de datos de Greenplum como origen

Los privilegios necesarios para la base de datos de Greenplum como origen se enumeran a continuación:

- CONNECT ON DATABASE *<database_name>*
- USAGE ON SCHEMA *<schema_name>*
- SELECT ON *<schema_name>.<table_name>*
- SELECT ON SEQUENCE *<schema_name>.<sequence_name>*

En el ejemplo anterior, sustituya los marcadores de posición del modo que sigue:

- Sustituya *database_name* por el nombre de la base de datos de origen.
- Sustituya *schema_name* por el nombre del esquema de origen.
- Sustituya *table_name* por el nombre de la tabla de origen.
- Sustituya *sequence_name* por el nombre de la secuencia.

Conectar a la base de datos de Greenplum como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de origen de Greenplum con AWS SCT.

Para conectarse a una base de datos Greenplum de origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.

2. Elija SAP ASE, y a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.

3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.

4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.

   - Para introducir manualmente las credenciales de la base de datos de origen de Greenplum, use las siguientes instrucciones:

Parámetro	Acción
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
Database	Escriba el nombre de la base de datos de Greenplum.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseñ a de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de

Parámetro	Acción
	AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Usar SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a la base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Verify Server Certificate: seleccione esta opción para verificar el certificado del servidor mediante un almacén de confianza. • Trust store: la ubicación de un almacén de confianza que contenga certificados.
Store password	AWS SCT crea un almacén seguro para almacenar certifica dos SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.
Greenplum Database driver path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Configuración de conversión de Greenplum a Amazon Redshift

Para editar la configuración de conversión de Greenplum a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Greenplum y, a continuación, elija Greenplum — Amazon Redshift. AWS SCT muestra todos los ajustes disponibles para la conversión de Greenplum a Amazon Redshift.

La configuración de conversión de Greenplum a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.
- Establecer el número máximo de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift de destino.

  En El número máximo de tablas para el clúster de Amazon Redshift de destino, elija el número de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift.

  Amazon Redshift tiene cuotas que limitan las tablas de uso para los distintos tipos de nodos de clúster. Si elige Auto, AWS SCT determina el número de tablas que se van a aplicar al clúster de Amazon Redshift de destino en función del tipo de nodo. Si lo desea, elija el valor manualmente. Para obtener más información, consulte Cuotas y límites de Amazon Redshift en la Guía de administración de Amazon Redshift.

  AWS SCT convierte todas las tablas de origen, aunque sean más de las que su clúster de Amazon Redshift puede almacenar. AWS SCTalmacena el código convertido en su proyecto y no lo aplica a la base de datos de destino. Si alcanza la cuota del clúster de Amazon Redshift para las tablas al aplicar el código convertido, AWS SCT mostrará un mensaje de advertencia. Además, AWS SCT aplica tablas a su clúster de Amazon Redshift de destino hasta que el número de tablas alcanza el límite.

- Migrar particiones de la tabla de origen a tablas independientes en Amazon Redshift. Para ello, seleccione Usar la vista UNION ALL e introduzca el número máximo de tablas de destino que AWS SCT puede crear para una sola tabla de origen.

  Amazon Redshift no admite particionamiento de tablas. Para simular este comportamiento y agilizar la ejecución de las consultas, AWS SCT puede migrar cada partición de la tabla de origen a una tabla independiente en Amazon Redshift. A continuación, AWS SCT crea una vista que incluye los datos de todas estas tablas.

  AWS SCT determina automáticamente el número de particiones de la tabla de origen. Según el tipo de particionamiento de la tabla de origen, este número puede superar la cuota de tablas que puede aplicar a su clúster de Amazon Redshift. Para evitar alcanzar esta cuota, introduzca el número máximo de tablas de destino que AWS SCT puede crear para las particiones de una sola tabla de origen. La opción predeterminada es 368 tablas, lo que representa una partición para 366 días de un año y dos tablas para las particiones `NO RANGE` y `UNKNOWN`.
- Aplicar compresión a las columnas de la tabla de Amazon Redshift. Para ello, seleccione Usar codificación de compresión.

  AWS SCT asigna automáticamente la codificación de compresión a las columnas mediante el algoritmo Amazon Redshift predeterminado. Para obtener más información, consulte Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  De forma predeterminada, Amazon Redshift no aplica compresión a las columnas definidas como claves de clasificación y distribución. Puede cambiar este comportamiento y aplicar compresión a estas columnas. Para ello, seleccione Usar codificación de compresión para columnas KEY. Puede seleccionar esta opción solo si selecciona la opción Usar codificación de compresión.

Configuración de optimización de la conversión de Greenplum a Amazon Redshift

Para editar la configuración de optimización de la conversión de Greenplum a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Greenplum y, a continuación, Greenplum — Amazon Redshift. En el panel izquierdo, elija Estrategias de optimización. AWS SCT muestra la configuración de optimización de la conversión para la conversión de Greenplum a Amazon Redshift.

La configuración de optimización de la conversión de Greenplum a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Trabajar con optimización automática de tablas. Para ello, seleccione Utilizar el ajuste automático de tablas de Amazon Redshift.

  La optimización automática de tablas es una capacidad de autoajuste presente en Amazon Redshift que optimiza automáticamente el diseño de tablas. Para obtener más información, consulte Trabajar con optimización automática de tablas en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  Para confiar únicamente en la optimización automática de las tablas, elija Ninguna en Estrategia inicial de selección de claves.

- Elegir las claves de clasificación y distribución según su estrategia.

  Puede elegir las claves de clasificación y distribución utilizando metadatos de Amazon Redshift, información estadística o ambas opciones. En estrategia inicial de selección de claves, en la pestaña Estrategias de optimización, elija una de las siguientes opciones:

  - Utilizar metadatos e ignorar la información estadística
  - Ignorar los metadatos, utilizar información estadística
  - Utilizar metadatos e información estadística

  En función de la opción que elija, puede seleccionar estrategias de optimización. A continuación, introduzca el valor (0—100) para cada estrategia. Estos valores definen la ponderación de cada estrategia. Con estos valores de ponderación, AWS SCT define cómo influye cada regla en la elección de las claves de distribución y clasificación. Los valores predeterminados se basan en las prácticas recomendadas de migración de AWS.

  Puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para definirla como una tabla pequeña. AWS SCT aplica el estilo de distribución ALL a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

- Configurar los detalles de la estrategia.

  Además de definir la ponderación de cada estrategia de optimización, puede configurar los ajustes de optimización. Para ello, elija Optimización de la conversión.

  - En Límite de columnas de clave de clasificación, introduzca el número máximo de columnas en la clave de clasificación.

- En Valor de umbral descompensado, introduzca el porcentaje (0—100) de un valor descompensado para una columna. AWS SCT excluye de la lista de candidatos a la clave de distribución las columnas con un valor de sesgo superior al umbral. AWS SCT define el valor descompensado de una columna como la relación porcentual entre el número de apariciones del valor más común y el número total de registros.
- En Consultas N principales de la tabla del historial de consultas, introduzca el número (1–100) de las consultas que se van a analizar con más frecuencia.
- En Seleccionar usuario de estadísticas, elija el usuario de la base de datos cuyas estadísticas de consulta desee analizar.

Además, en la pestaña Estrategias de optimización, puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para considerarla una tabla pequeña. AWS SCT aplica el estilo de distribución ALL a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

Utilizar Netezza como origen para AWS SCT

Puede utilizar AWS SCT para convertir esquemas, objetos de código y código de aplicación de Netezza a Amazon Redshift.

Privilegios para Netezza como origen

Los privilegios necesarios para Netezza como origen se enumeran a continuación:

- select on system.definition_schema.system view
- select on system.definition_schema.system table
- select on system.definition_schema.management table
- list on *<database_name>*
- list on *<schema_name>*
- list on *<database_name>*.all.table
- list on *<database_name>*.all.external table
- list on *<database_name>*.all.view
- list on *<database_name>*.all.materialized view
- list on *<database_name>*.all.procedure

- list on *<database_name>*.all.sequence
- list on *<database_name>*.all.function
- list on *<database_name>*.all.aggregate

En el ejemplo anterior, sustituya los marcadores de posición del modo que sigue:

- Sustituya *database_name* por el nombre de la base de datos de origen.
- Sustituya *schema_name* por el nombre del esquema de origen.

AWS SCT requiere acceso a las siguientes vistas y tablas del sistema. Puede conceder el acceso a estos objetos en lugar de conceder el acceso a `system.definition_schema.system view` y `system.definition_schema.system tables` en la lista anterior.

- select on system.definition_schema._t_aggregate
- select on system.definition_schema._t_class
- select on system.definition_schema._t_constraint
- select on system.definition_schema._t_const_relattr
- select on system.definition_schema._t_database
- select on system.definition_schema._t_grpobj_priv
- select on system.definition_schema._t_grpusr
- select on system.definition_schema._t_hist_config
- select on system.definition_schema._t_object
- select on system.definition_schema._t_object_classes
- select on system.definition_schema._t_proc
- select on system.definition_schema._t_type
- select on system.definition_schema._t_user
- select on system.definition_schema._t_usrobj_priv
- select on system.definition_schema._vt_sequence
- select on system.definition_schema._v_aggregate
- select on system.definition_schema._v_constraint_depends
- select on system.definition_schema._v_database
- select on system.definition_schema._v_datatype

- select on system.definition_schema._v_dslice
- select on system.definition_schema._v_function
- select on system.definition_schema._v_group
- select on system.definition_schema._v_obj_relation
- select on system.definition_schema._v_obj_relation_xdb
- select on system.definition_schema._v_procedure
- select on system.definition_schema._v_relation_column
- select on system.definition_schema._v_relation_keydata
- select on system.definition_schema._v_relobjclasses
- select on system.definition_schema._v_schema_xdb
- select on system.definition_schema._v_sequence
- select on system.definition_schema._v_synonym
- select on system.definition_schema._v_system_info
- select on system.definition_schema._v_sys_constraint
- select on system.definition_schema._v_sys_object_dslice_info
- select on system.definition_schema._v_sys_user
- select on system.definition_schema._v_table
- select on system.definition_schema._v_sys_constraint
- select on system.definition_schema._v_table
- select on system.definition_schema._v_table
- select on system.definition_schema._v_table
- select on system.definition_schema._v_user
- select on system.definition_schema._v_user
- select on system.information_schema._v_relation_column
- select on system.information_schema._v_table
- select on $hist_column_access_*

Conectar a Netezza como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de origen de Netezza con AWS Schema Conversion Tool.

Para conectarse a una base de datos Netezza de origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Seleccione Netezza y, a continuación, seleccione Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión a la base de datos de origen de Netezza, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseñ

Parámetro	Acción
	a de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Store password	AWS SCT crea un almacén seguro para almacenar certifica dos SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.
Netezza driver path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Configurar la replicación continua de datos

Tras convertir los esquemas de base de datos de Netezza y aplicarlos a la base de datos de Amazon Redshift, puede migrar los datos con agentes de extracción de datos de AWS SCT. El agente extrae los datos y los carga en el bucket de Amazon S3. Puede utilizar AWS SCT para copiar los datos de Amazon S3 a Amazon Redshift.

Si los datos de la base de datos de origen cambian durante el proceso de migración, puede capturar los cambios en curso con sus agentes de extracción de datos de AWS SCT. Seguidamente, puede

replicar estos cambios continuos en la base de datos de destino después de completar la migración de datos inicial. Este proceso se denomina replicación continua o captura de datos de cambio (CDC).

Para configurar la replicación continua de datos para migraciones de Netezza a Amazon Redshift

1. En la base de datos de origen, cree una base de datos de historial. Puede utilizar el siguiente ejemplo de código en la interfaz de la línea de comandos (CLI) de Netezza.

   ```
   nzhistcreatedb -d history_database_name -t query -v 1 -u load_user -o histdb_owner
    -p your_password
   ```

   En el ejemplo anterior, sustituya *history_database_name* por el nombre de la base de datos de historial. A continuación, sustituya *load_user* por el nombre del usuario que ha definido para cargar los datos del historial en la base de datos. A continuación, sustituya *histdb_owner* por el nombre del usuario que ha definido como propietario de la base de datos del historial. Asegúrese de haber creado este usuario y de haberle concedido el permiso `CREATE DATABASE`. Por último, sustituya *your_password* por una contraseña segura.

2. Configure el registro del historial. Para ello, use el siguiente ejemplo de código.

   ```
   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;
   ```

   En el ejemplo anterior, sustituya *history_configuration_name* y *history_database_name* por los nombres de la configuración del historial y de la base de datos del historial. A continuación, sustituya *load_user* por el nombre del usuario que ha definido para cargar los datos del historial en la base de datos. Por último, sustituya *your_password* por una contraseña segura.

3. Otorgar permisos de lectura para todas las tablas de la base de datos del historial. Puede utilizar el siguiente código de ejemplo para conceder el permiso `SELECT`.

   ```
   GRANT SELECT ON history_database_name.ALL.TABLE TO your_user;
   ```

   En el ejemplo anterior, sustituya *history_database_name* por el nombre de la base de datos de historial. A continuación, sustituya *your_user* por el nombre del usuario con permisos

mínimos para trabajar con la base de datos de Netezza. Utilice las credenciales de este usuario de la base de datos en AWS SCT.

4. Recopile las estadísticas de cada tabla del esquema de origen para obtener información sobre la cardinalidad de las columnas. Puede utilizar el siguiente comando para generar estadísticas en la base de datos de historial.

```
GENERATE STATISTICS on "schema_name"."table_name";
```

En el ejemplo anterior, sustituya *schema_name* y *table_name* por el nombre del esquema y la tabla de la base de datos.

5. Asegúrese de que ha completado los requisitos. Para ello, ejecute la siguiente consulta:

```
SELECT COUNT(*)
 FROM history_database_name.history_schema_name."$hist_column_access_N";
```

En el ejemplo anterior, sustituya *history_database_name* y *history_schema_name* por el nombre del esquema y la base de datos del historial. A continuación, sustituya *N* por el número de versión de la base de datos del historial. Para obtener más información sobre las versiones de la base de dato del historial, consulte Documentación de IBM Netezza.

6. Instale el agente de extracción de datos. Para obtener más información, consulte Instalación de agentes de extracción.

   Asegúrese de que el parámetro `{working.folder}` del archivo `settings.properties` de todas las instancias del extractor apunte a la misma carpeta. En este caso, sus extractores pueden coordinar la sesión de CDC y utilizar un único punto de transacción para todas las tareas secundarias.

7. Registre su agente de extracción de datos. Para obtener más información, consulte Registrar los agentes de extracción con el AWS Schema Conversion Tool.

8. Cree su tarea de CDC. Para obtener más información, consulte Crear, ejecutar y supervisar una AWS SCT tarea.

   a. Abra su proyecto en AWS SCT. En el panel izquierdo, elija la tabla de origen. Abra el menú contextual (clic secundario) y seleccione Crear tarea local.

   b. En Nombre de la tarea, introduzca un nombre para la tarea de migración de datos.

   c. En Modo de migración, seleccione Extraer, cargar y copiar.

   d. Seleccione Habilitar CDC.

e. Elija la pestaña Configuración de CDC y defina el ámbito y la programación de las sesiones de CDC.

f. Seleccione Probar tarea para comprobar que se puede conectar a su carpeta de trabajo, bucket de Amazon S3 y almacenamiento de datos de Amazon Redshift.

g. Seleccione Crear para crear su tarea.

h. Elija la pestaña Tareas, seleccione su tarea de la lista y elija Iniciar.

9. La tarea de AWS SCT mantiene la coherencia transaccional en la base de datos de destino. El agente de extracción de datos replica las transacciones del origen en el orden de los identificadores de transacción.

   Si detiene alguna de las sesiones de migración o si se produce un error, el procesamiento de la CDC también se detiene.

Configuración de conversión de Netezza a Amazon Redshift

Para editar la configuración de conversión de Netezza a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Netezza y, a continuación, elija Netezza — Amazon Redshift. AWS SCT muestra todos los ajustes disponibles para la conversión de Netezza a Amazon Redshift.

La configuración de conversión de Netezza a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.

- Establecer el número máximo de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift de destino.

En El número máximo de tablas para el clúster de Amazon Redshift de destino, elija el número de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift.

Amazon Redshift tiene cuotas que limitan las tablas de uso para los distintos tipos de nodos de clúster. Si elige Auto, AWS SCT determina el número de tablas que se van a aplicar al clúster de Amazon Redshift de destino en función del tipo de nodo. Si lo desea, elija el valor manualmente. Para obtener más información, consulte Cuotas y límites de Amazon Redshift en la Guía de administración de Amazon Redshift.

AWS SCT convierte todas las tablas de origen, aunque sean más de las que su clúster de Amazon Redshift puede almacenar. AWS SCTalmacena el código convertido en su proyecto y no lo aplica a la base de datos de destino. Si alcanza la cuota del clúster de Amazon Redshift para las tablas al aplicar el código convertido, AWS SCT mostrará un mensaje de advertencia. Además, AWS SCT aplica tablas a su clúster de Amazon Redshift de destino hasta que el número de tablas alcanza el límite.

- Aplicar compresión a las columnas de la tabla de Amazon Redshift. Para ello, seleccione Usar codificación de compresión.

  AWS SCT asigna automáticamente la codificación de compresión a las columnas mediante el algoritmo Amazon Redshift predeterminado. Para obtener más información, consulte Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  De forma predeterminada, Amazon Redshift no aplica compresión a las columnas definidas como claves de clasificación y distribución. Puede cambiar este comportamiento y aplicar compresión a estas columnas. Para ello, seleccione Usar codificación de compresión para columnas KEY. Puede seleccionar esta opción solo si selecciona la opción Usar codificación de compresión.

Configuración de optimización de la conversión de Netezza a Amazon Redshift

Para editar la configuración de optimización de la conversión de Netezza a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Netezza y, a continuación, Netezza — Amazon Redshift. En el panel izquierdo, elija Estrategias de optimización. AWS SCT muestra la configuración de optimización de la conversión para la conversión de Netezza a Amazon Redshift.

La configuración de optimización de la conversión de Netezza a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Trabajar con optimización automática de tablas. Para ello, seleccione Utilizar el ajuste automático de tablas de Amazon Redshift.

  La optimización automática de tablas es una capacidad de autoajuste presente en Amazon Redshift que optimiza automáticamente el diseño de tablas. Para obtener más información, consulte Trabajar con optimización automática de tablas en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  Para confiar únicamente en la optimización automática de las tablas, elija Ninguna en Estrategia inicial de selección de claves.
- Elegir las claves de clasificación y distribución según su estrategia.

  Puede elegir las claves de clasificación y distribución utilizando metadatos de Amazon Redshift, información estadística o ambas opciones. En estrategia inicial de selección de claves, en la pestaña Estrategias de optimización, elija una de las siguientes opciones:

  - Utilizar metadatos e ignorar la información estadística
  - Ignorar los metadatos, utilizar información estadística
  - Utilizar metadatos e información estadística

  En función de la opción que elija, puede seleccionar estrategias de optimización. A continuación, introduzca el valor (0—100) para cada estrategia. Estos valores definen la ponderación de cada estrategia. Con estos valores de ponderación, AWS SCT define cómo influye cada regla en la elección de las claves de distribución y clasificación. Los valores predeterminados se basan en las prácticas recomendadas de migración de AWS.

  Puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para definirla como una tabla pequeña. AWS SCT aplica el estilo de distribución ALL a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.
- Configurar los detalles de la estrategia.

  Además de definir la ponderación de cada estrategia de optimización, puede configurar los ajustes de optimización. Para ello, elija Optimización de la conversión.

- En Límite de columnas de clave de clasificación, introduzca el número máximo de columnas en la clave de clasificación.
- En Valor de umbral descompensado, introduzca el porcentaje (0—100) de un valor descompensado para una columna. AWS SCT excluye de la lista de candidatos a la clave de distribución las columnas con un valor de sesgo superior al umbral. AWS SCT define el valor descompensado de una columna como la relación porcentual entre el número de apariciones del valor más común y el número total de registros.
- En Consultas N principales de la tabla del historial de consultas, introduzca el número (1–100) de las consultas que se van a analizar con más frecuencia.
- En Seleccionar usuario de estadísticas, elija el usuario de la base de datos cuyas estadísticas de consulta desee analizar.

Además, en la pestaña Estrategias de optimización, puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para considerarla una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

Utilizar Oracle Data Warehouse como origen para AWS SCT

Puede utilizar AWS SCT para convertir esquemas, objetos de código y código de aplicaciones de Oracle Data Warehouse a Amazon Redshift o Amazon Redshift y AWS Glue utilizados en combinación.

Privilegios para Oracle Data Warehouse como origen

Los privilegios necesarios para Oracle Data Warehouse como origen se enumeran a continuación:

- connect
- select_catalog_role
- select any dictionary

Conectar a Oracle Data Warehouse como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de Oracle Data Warehouse de origen con AWS Schema Conversion Tool.

Para conectarse a una base de datos de Oracle Data Warehouse de origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Elija Oracle y, a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión al almacenamiento de datos de origen de Oracle, siga las siguientes instrucciones:

Parámetro	Acción

Type	Seleccione el tipo de conexión a la base de datos. En función de su tipo, proporcione la siguiente información adicional:
	• SID
	  • Server name: escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
	  • Server port: el puerto que utilice para conectarse al servidor de su base de datos de origen.
	  • Oracle SID: el ID del sistema (SID) de Oracle. Para encontrar el SID de Oracle, envíe la siguiente consulta a su base de datos de Oracle:

Parámetro	Acción

	`SELECT sys_context('userenv','instance_name') AS SID FROM dual;`
	• Service Name
	  • Server name: el nombre DNS o la dirección IP de su servidor de base de datos de origen.
	  • Server port: el puerto que utilice para conectarse al servidor de su base de datos de origen.
	  • Service Name: el nombre del servicio de Oracle al que conectarse.
	• TNS alias
	  • TNS file path: la ruta al archivo que contiene la información de conexión del nombre Transparent Network Substrate (TNS).
	  • TNS file path: el alias TNS de este archivo que utilizar para conectarse a la base de datos de origen.
	• TNS connect identifier
	  • TNS connect identifier: identificador de la información de conexión del TNS registrado.

User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.

Parámetro	Acción
Usar SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a la base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • SSL authentication: seleccione esta opción para utilizar autenticación SSL para la conexión. • Trust store: la ubicación de un almacén de confianza que contenga certificados. • Key store: la ubicación de un almacén de claves que contenga una clave privada y certificados. Este valor es obligatorio si está seleccionado SSL authentication. De lo contrario, es opcional.
Store password	AWS SCT crea un almacén seguro para almacenar certifica dos SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.
Oracle driver path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Configuración de conversión de Oracle Data Warehouse a Amazon Redshift

Para editar la configuración de conversión de Oracle Data Warehouse a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Oracle y, a continuación, elija Oracle — Amazon Redshift. AWS SCT muestra todos los ajustes disponibles para la conversión de Oracle Data Warehouse a Amazon Redshift.

La configuración de conversión de Oracle Data Warehouse a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.
- Establecer el número máximo de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift de destino.

  En El número máximo de tablas para el clúster de Amazon Redshift de destino, elija el número de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift.

  Amazon Redshift tiene cuotas que limitan las tablas de uso para los distintos tipos de nodos de clúster. Si elige Auto, AWS SCT determina el número de tablas que se van a aplicar al clúster de Amazon Redshift de destino en función del tipo de nodo. Si lo desea, elija el valor manualmente. Para obtener más información, consulte Cuotas y límites de Amazon Redshift en la Guía de administración de Amazon Redshift.

  AWS SCT convierte todas las tablas de origen, aunque sean más de las que su clúster de Amazon Redshift puede almacenar. AWS SCTalmacena el código convertido en su proyecto y no lo aplica a la base de datos de destino. Si alcanza la cuota del clúster de Amazon Redshift para las tablas al aplicar el código convertido, AWS SCT mostrará un mensaje de advertencia. Además, AWS SCT aplica tablas a su clúster de Amazon Redshift de destino hasta que el número de tablas alcanza el límite.

- Migrar particiones de la tabla de origen a tablas independientes en Amazon Redshift. Para ello, seleccione Usar la vista UNION ALL e introduzca el número máximo de tablas de destino que AWS SCT puede crear para una sola tabla de origen.

  Amazon Redshift no admite particionamiento de tablas. Para simular este comportamiento y agilizar la ejecución de las consultas, AWS SCT puede migrar cada partición de la tabla de origen a una tabla independiente en Amazon Redshift. A continuación, AWS SCT crea una vista que incluye los datos de todas estas tablas.

  AWS SCT determina automáticamente el número de particiones de la tabla de origen. Según el tipo de particionamiento de la tabla de origen, este número puede superar la cuota de tablas que puede aplicar a su clúster de Amazon Redshift. Para evitar alcanzar esta cuota, introduzca el número máximo de tablas de destino que AWS SCT puede crear para las particiones de una sola tabla de origen. La opción predeterminada es 368 tablas, lo que representa una partición para 366 días de un año y dos tablas para las particiones `NO RANGE` y `UNKNOWN`.
- Convertir funciones de formato de tipos de datos como `TO_CHAR`, `TO_DATE` y `TO_NUMBER` con elementos de formato de fecha y hora que Amazon Redshift no admite. De forma predeterminada, AWS SCT utiliza las funciones del paquete de extensión para simular el uso de estos elementos de formato no compatibles en el código convertido.

  El modelo de formato de fecha y hora de Oracle incluye más elementos en comparación con las cadenas de formato de fecha y hora de Amazon Redshift. Si su código fuente incluye únicamente elementos de formato de fecha y hora compatibles con Amazon Redshift, no necesitará las funciones del paquete de extensión incluidas en el código convertido. Para evitar utilizar las funciones del paquete de extensión en el código convertido, seleccione Los elementos de formato Datetype que utiliza en el código de Oracle son similares a las cadenas de formato de fecha y hora de Amazon Redshift. En este caso, el código convertido funciona más rápido.

  El modelo de formato numérico de Oracle incluye más elementos en comparación con las cadenas de formato numérico de Amazon Redshift. Si su código fuente incluye únicamente elementos de formato numérico compatibles con Amazon Redshift, no necesitará las funciones del paquete de extensión incluidas en el código convertido. Para evitar utilizar las funciones del paquete de extensión en el código convertido, seleccione Los elementos de formato numérico que utiliza en el código de Oracle son similares a las cadenas de formato numérico de Amazon Redshift. En este caso, el código convertido funciona más rápido.
- Convertir funciones analíticas `LEAD` y `LAG` de Oracle. De forma predeterminada, AWS SCT genera un elemento de acción para cada función `LEAD` y `LAG`.

Si su código fuente no utiliza los valores predeterminados de compensación en estas funciones, AWS SCT puede simular el uso de estas funciones con la función NVL. Para ello, seleccione Utilizar la función NVL para simular el comportamiento de las funciones LEAD y LAG de Oracle.

- Para simular el comportamiento de las claves principales y únicas del clúster de Amazon Redshift, seleccione Simular el comportamiento de claves principales y únicas.

  Amazon Redshift no exige claves únicas y principales y las utiliza únicamente con fines informativos. Si utiliza estas restricciones en el código, asegúrese de que AWS SCT simula su comportamiento en el código convertido.

- Aplicar compresión a las columnas de la tabla de Amazon Redshift. Para ello, seleccione Usar codificación de compresión.

  AWS SCT asigna automáticamente la codificación de compresión a las columnas mediante el algoritmo Amazon Redshift predeterminado. Para obtener más información, consulte Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  De forma predeterminada, Amazon Redshift no aplica compresión a las columnas definidas como claves de clasificación y distribución. Puede cambiar este comportamiento y aplicar compresión a estas columnas. Para ello, seleccione Usar codificación de compresión para columnas KEY. Puede seleccionar esta opción solo si selecciona la opción Usar codificación de compresión.

Configuración de optimización de la conversión de Oracle Data Warehouse a Amazon Redshift

Para editar la configuración de optimización de la conversión de Oracle Data Warehouse a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Oracle y, a continuación, Oracle — Amazon Redshift. En el panel izquierdo, elija Estrategias de optimización. AWS SCT muestra la configuración de optimización de la conversión para la conversión de Oracle Data Warehouse a Amazon Redshift.

La configuración de optimización de la conversión de Oracle Data Warehouse a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Trabajar con optimización automática de tablas. Para ello, seleccione Utilizar el ajuste automático de tablas de Amazon Redshift.

La optimización automática de tablas es una capacidad de autoajuste presente en Amazon Redshift que optimiza automáticamente el diseño de tablas. Para obtener más información, consulte Trabajar con optimización automática de tablas en la Guía para desarrolladores de bases de datos de Amazon Redshift.

Para confiar únicamente en la optimización automática de las tablas, elija Ninguna en Estrategia inicial de selección de claves.

- Elegir las claves de clasificación y distribución según su estrategia.

  Puede elegir las claves de clasificación y distribución utilizando metadatos de Amazon Redshift, información estadística o ambas opciones. En estrategia inicial de selección de claves, en la pestaña Estrategias de optimización, elija una de las siguientes opciones:

  - Utilizar metadatos e ignorar la información estadística
  - Ignorar los metadatos, utilizar información estadística
  - Utilizar metadatos e información estadística

  En función de la opción que elija, puede seleccionar estrategias de optimización. A continuación, introduzca el valor (0—100) para cada estrategia. Estos valores definen la ponderación de cada estrategia. Con estos valores de ponderación, AWS SCT define cómo influye cada regla en la elección de las claves de distribución y clasificación. Los valores predeterminados se basan en las prácticas recomendadas de migración de AWS.

  Puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para definirla como una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

- Configurar los detalles de la estrategia.

  Además de definir la ponderación de cada estrategia de optimización, puede configurar los ajustes de optimización. Para ello, elija Optimización de la conversión.

  - En Límite de columnas de clave de clasificación, introduzca el número máximo de columnas en la clave de clasificación.
  - En Valor de umbral descompensado, introduzca el porcentaje (0—100) de un valor descompensado para una columna. AWS SCT excluye de la lista de candidatos a la clave de distribución las columnas con un valor de sesgo superior al umbral. AWS SCT define el valor

descompensado de una columna como la relación porcentual entre el número de apariciones del valor más común y el número total de registros.

- En Consultas N principales de la tabla del historial de consultas, introduzca el número (1–100) de las consultas que se van a analizar con más frecuencia.
- En Seleccionar usuario de estadísticas, elija el usuario de la base de datos cuyas estadísticas de consulta desee analizar.

Además, en la pestaña Estrategias de optimización, puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para considerarla una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

Utilizar Snowflake como origen para AWS SCT

Puede utilizar AWS SCT para convertir esquemas, objetos de código y código de aplicación de Snowflake a Amazon Redshift.

Privilegios para Snowflake como base de datos de origen

Puede crear un rol con privilegios y asignar a este rol el nombre de un usuario utilizando el rol `SECURITYADMIN` y el contexto de la sesión de `SECURITYADMIN`.

El siguiente ejemplo crea privilegios mínimos y se los concede al usuario `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;
```

En el ejemplo anterior, sustituya los marcadores de posición del modo que sigue:

- Sustituya *role_name* por el nombre de un rol con privilegios de solo lectura.
- Sustituya *db_name* por el nombre de la base de datos de origen.
- Sustituya *schema_name* por el nombre del esquema de origen.
- Sustituya *datawarehousename* por el nombre de un almacenamiento de datos necesario.
- Sustituya *min_privs* por el nombre de un usuario que tenga privilegios mínimos.

Los parámetros `DEFAULT_ROLE` y `DEFAULT_WAREHOUSE` son sensibles a las claves.

Configurar el acceso seguro a Amazon S3

Las políticas de seguridad y administración de acceso de un bucket de Amazon S3 permiten a Snowflake acceder, leer y escribir datos en el bucket de S3. Puede configurar el acceso seguro a un bucket privado de Amazon S3 mediante el tipo de objeto `STORAGE INTEGRATION` de Snowflake. Un objeto de integración de almacenamiento de Snowflake delega la responsabilidad de autenticación en una entidad de administración de identidades y accesos de Snowflake.

Para obtener más información, consulte Configurar una integración de almacenamiento de Snowflake para acceder a Amazon S3 en la documentación de Snowflake.

Conectar a Snowflake como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de origen con AWS Schema Conversion Tool.

Para conectarse a una base de datos de origen de Snowflake

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Seleccione Snowflake y, a continuación, seleccione Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.

3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.

   - Para introducir manualmente la información de conexión del almacenamiento de datos de origen de Snowflake, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
Database	Escriba el nombre de la base de datos de Snowflake.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT almacena la contraseña en formato cifrado solo si lo solicita explícitamente.
Use SSL	Seleccione esta opción si desea utilizar capa de sockets seguros (SSL) para conectarse a la base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Private key path: la ubicación de una clave privada.

Parámetro	Acción
	• Passphrase: la contraseña de la clave privada. Para obtener más información sobre la compatibilidad con SSL de Snowflake, consulte Configurar opciones de seguridad para las conexiones.
Store password	AWS SCT crea un almacén seguro para almacenar certifica dos SSL y contraseñas de la base de datos. Si configura esta opción, puede almacenar la contraseña de la base de datos. De este modo, podrá conectarse rápidamente a la base de datos sin tener que introducir la contraseña.
Snowflake driver path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Limitaciones de Snowflake como origen

A continuación, se describen las limitaciones que se aplican cuando se utiliza Snowflake como origen para AWS SCT:

- Los identificadores de objeto deben ser únicos en el contexto del tipo de objeto y del objeto principal:

  Base de datos

  Los identificadores de esquema deben ser únicos dentro de una base de datos.

Esquemas

Los identificadores de objetos, como los de tablas y vistas, deben ser únicos dentro de un esquema.

Tablas/Vistas

Los identificadores de columna deben ser únicos dentro de una tabla.

- El número máximo de tablas para los tipos de nodos de clúster large y xlarge es de 9 900. El número máximo de tablas para los tipos de nodos de clúster 8xlarge es 100 000. El límite incluye tablas temporales, tanto las definidas por el usuario como las creadas con Amazon Redshift durante el procesamiento de las consultas o el mantenimiento del sistema. Para obtener más información, consulte Cuotas de Amazon Redshift en la Guía de administración de clústeres de Amazon Redshift.
- Para los procedimientos almacenados, el número máximo de argumentos de entrada y salida es 32.

Tipos de datos de origen para Snowflake

A continuación, encontrará los tipos de datos de origen de Snowflake que se admiten cuando se utiliza AWS SCT y la asignación predeterminada a un destino de Amazon Redshift.

Tipos de datos de Snowflake	Tipos de datos de Amazon Redshift
NUMBER	NUMERIC(38)
NUMBER(p)	Si p es =< 4, entonces SMALLINT Si p es => 5 y =< 9, entonces INTEGER Si p es => 10 y =< 18, entonces INTEGER Si p es => 19 entonces NUMERIC (p)
NUMBER(p, 0)	Si p es =< 4, entonces SMALLINT Si p es => 5 y =< 9, entonces INTEGER Si p es => 10 y =< 18, entonces INTEGER

Tipos de datos de Snowflake	Tipos de datos de Amazon Redshift
	Si p es => 19 entonces NUMERIC (p,0)
NUMBER(p, s)	Si p es => 1 y =< 38, y si s es => 1 y =< 37, entonces NUMERIC(p,s)
FLOAT	FLOAT
TEXT Caracteres Unicode de hasta 16 777 216 bytes; hasta 4 bytes por carácter.	VARCHAR(MAX)
TEXT(p) Caracteres Unicode de hasta 65 535 bytes; hasta 4 bytes por carácter.	Si p es =< 65 535, entonces, VARCHAR (p)
TEXT(p) Caracteres Unicode de hasta 16 777 216 bytes; hasta 4 bytes por carácter.	Si p es => 65 535 y =< 16 777 216, entonces, VARCHAR (MAX)
BINARY Caracteres de un solo byte de hasta 8 388 608 bytes; 1 byte por carácter.	VARCHAR(MAX)
BINARY(p) Caracteres de un solo byte de hasta 65 535 bytes; 1 byte por carácter.	VARCHAR(p)
BINARY(p) Caracteres de un solo byte de hasta 8 388 608 bytes; 1 byte por carácter.	VARCHAR(MAX)
BOOLEAN	BOOLEAN

Tipos de datos de Snowflake	Tipos de datos de Amazon Redshift
DATE	DATE
TIME Valores de hora entre 00:00:00 y 23:59:59,999999999.	VARCHAR(18)
TIME(f) Valores de hora entre 00:00:00 y 23:59:59,9(f).	VARCHAR(n) – 9 + dt-attr-1
TIMESTAMP_NTZ	TIMESTAMP
TIMESTAMP_TZ	TIMESTAMPTZ

Configuración de conversión de Snowflake a Amazon Redshift

Para editar la configuración de conversión de Snowflake a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Snowflake y, a continuación, elija Snowflake — Amazon Redshift. AWS SCT muestra todos los ajustes disponibles para la conversión de Snowflake a Amazon Redshift.

La configuración de conversión de Snowflake a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.

- Establecer el número máximo de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift de destino.

  En El número máximo de tablas para el clúster de Amazon Redshift de destino, elija el número de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift.

  Amazon Redshift tiene cuotas que limitan las tablas de uso para los distintos tipos de nodos de clúster. Si elige Auto, AWS SCT determina el número de tablas que se van a aplicar al clúster de Amazon Redshift de destino en función del tipo de nodo. Si lo desea, elija el valor manualmente. Para obtener más información, consulte Cuotas y límites de Amazon Redshift en la Guía de administración de Amazon Redshift.

  AWS SCT convierte todas las tablas de origen, aunque sean más de las que su clúster de Amazon Redshift puede almacenar. AWS SCTalmacena el código convertido en su proyecto y no lo aplica a la base de datos de destino. Si alcanza la cuota del clúster de Amazon Redshift para las tablas al aplicar el código convertido, AWS SCT mostrará un mensaje de advertencia. Además, AWS SCT aplica tablas a su clúster de Amazon Redshift de destino hasta que el número de tablas alcanza el límite.
- Aplicar compresión a las columnas de la tabla de Amazon Redshift. Para ello, seleccione Usar codificación de compresión.

  AWS SCT asigna automáticamente la codificación de compresión a las columnas mediante el algoritmo Amazon Redshift predeterminado. Para obtener más información, consulte Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  De forma predeterminada, Amazon Redshift no aplica compresión a las columnas definidas como claves de clasificación y distribución. Puede cambiar este comportamiento y aplicar compresión a estas columnas. Para ello, seleccione Usar codificación de compresión para columnas KEY. Puede seleccionar esta opción solo si selecciona la opción Usar codificación de compresión.

Configuración de optimización de la conversión de Snowflake a Amazon Redshift

Para editar la configuración de optimización de la conversión de Snowflake a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Snowflake y, a continuación, Snowflake — Amazon Redshift. En el panel izquierdo, elija Estrategias de optimización. AWS SCT muestra la configuración de optimización de la conversión para la conversión de Snowflake a Amazon Redshift.

La configuración de optimización de la conversión de Snowflake a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Trabajar con optimización automática de tablas Para ello, seleccione Utilizar el ajuste automático de tablas de Amazon Redshift.

  La optimización automática de tablas es una capacidad de autoajuste presente en Amazon Redshift que optimiza automáticamente el diseño de tablas. Para obtener más información, consulte Trabajar con optimización automática de tablas en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  Para confiar únicamente en la optimización automática de las tablas, elija Ninguna en Estrategia inicial de selección de claves.
- Elegir las claves de clasificación y distribución según su estrategia.

  Puede elegir las claves de clasificación y distribución utilizando metadatos de Amazon Redshift, información estadística o ambas opciones. En estrategia inicial de selección de claves, en la pestaña Estrategias de optimización, elija una de las siguientes opciones:

  - Utilizar metadatos e ignorar la información estadística
  - Ignorar los metadatos, utilizar información estadística
  - Utilizar metadatos e información estadística

  En función de la opción que elija, puede seleccionar estrategias de optimización. A continuación, introduzca el valor (0—100) para cada estrategia. Estos valores definen la ponderación de cada estrategia. Con estos valores de ponderación, AWS SCT define cómo influye cada regla en la elección de las claves de distribución y clasificación. Los valores predeterminados se basan en las prácticas recomendadas de migración de AWS.

  Puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para definirla como una tabla pequeña. AWS SCT aplica el estilo de distribución ALL a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.
- Configurar los detalles de la estrategia.

  Además de definir la ponderación de cada estrategia de optimización, puede configurar los ajustes de optimización. Para ello, elija Optimización de la conversión.

- En Límite de columnas de clave de clasificación, introduzca el número máximo de columnas en la clave de clasificación.
- En Valor de umbral descompensado, introduzca el porcentaje (0—100) de un valor descompensado para una columna. AWS SCT excluye de la lista de candidatos a la clave de distribución las columnas con un valor de sesgo superior al umbral. AWS SCT define el valor descompensado de una columna como la relación porcentual entre el número de apariciones del valor más común y el número total de registros.
- En Consultas N principales de la tabla del historial de consultas, introduzca el número (1–100) de las consultas que se van a analizar con más frecuencia.
- En Seleccionar usuario de estadísticas, elija el usuario de la base de datos cuyas estadísticas de consulta desee analizar.

Además, en la pestaña Estrategias de optimización, puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para considerarla una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

Utilizar Microsoft SQL Server Data Warehouse como origen para AWS SCT

Puede utilizar AWS SCT para convertir esquemas, objetos de código y código de aplicación de Microsoft SQL Server DW a Amazon Redshift o Amazon Redshift y AWS Glue utilizados en combinación.

Privilegios para Microsoft SQL Server Data Warehouse como origen

Los privilegios necesarios para Microsoft SQL Server Data Warehouse como origen se enumeran a continuación:

- VIEW DEFINITION
- VIEW DATABASE STATE
- SELECT ON SCHEMA :: *<schema_name>*

En el ejemplo anterior, sustituya el marcador de posición *<schema_name>* por el nombre del source_schema de origen.

Repita la concesión para cada base de datos cuyo esquema vaya a convertir.

Además, conceda el siguiente privilegio, y ejecute la concesión en la base de datos maestra:

- VIEW SERVER STATE

Limitaciones en el uso de SQL Server Data Warehouse como origen

Actualmente no se admite el uso de Microsoft SQL Server Parallel Data Warehouse (PDW) como origen.

Conectar a SQL Server Data Warehouse como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de origen de SQL Server Data Warehouse con AWS Schema Conversion Tool.

Para conectarse a una base de datos de origen de Microsoft SQL Server Data Warehouse

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Elija Microsoft SQL Server y, a continuación, elija Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión al almacenamiento de datos de origen de Microsoft SQL Server, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del servicio de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
Instance name	Escriba el nombre de la instancia del almacenamiento de datos de SQL Server.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseñ a de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Use SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a la base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Trust server certificate: seleccione esta opción para confiar en el certificado del servidor. • Trust store: un almacén de confianza que haya configura do en Configuración global.

Parámetro	Acción
Store password	AWS SCT crea un almacén seguro para almacenar certificados SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.
SQL Server driver path	Escriba la ruta al controlador que va a usar para conectarse a la base de datos de origen. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más información, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Configuración de conversión de SQL Server Data Warehouse a Amazon Redshift

Para editar la configuración de conversión de SQL Server Data Warehouse a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Microsoft SQL Server y, a continuación, elija Microsoft SQL Server — Amazon Redshift. AWS SCT muestra todos los ajustes disponibles para la conversión de SQl Server Data Warehouse a Amazon Redshift.

La configuración de conversión de SQL Server Data Warehouse a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade

comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.

- Establecer el número máximo de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift de destino.

  En El número máximo de tablas para el clúster de Amazon Redshift de destino, elija el número de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift.

  Amazon Redshift tiene cuotas que limitan las tablas de uso para los distintos tipos de nodos de clúster. Si elige Auto, AWS SCT determina el número de tablas que se van a aplicar al clúster de Amazon Redshift de destino en función del tipo de nodo. Si lo desea, elija el valor manualmente. Para obtener más información, consulte Cuotas y límites de Amazon Redshift en la Guía de administración de Amazon Redshift.

  AWS SCT convierte todas las tablas de origen, aunque sean más de las que su clúster de Amazon Redshift puede almacenar. AWS SCTalmacena el código convertido en su proyecto y no lo aplica a la base de datos de destino. Si alcanza la cuota del clúster de Amazon Redshift para las tablas al aplicar el código convertido, AWS SCT mostrará un mensaje de advertencia. Además, AWS SCT aplica tablas a su clúster de Amazon Redshift de destino hasta que el número de tablas alcanza el límite.
- Migrar particiones de la tabla de origen a tablas independientes en Amazon Redshift. Para ello, seleccione Usar la vista UNION ALL e introduzca el número máximo de tablas de destino que AWS SCT puede crear para una sola tabla de origen.

  Amazon Redshift no admite particionamiento de tablas. Para simular este comportamiento y agilizar la ejecución de las consultas, AWS SCT puede migrar cada partición de la tabla de origen a una tabla independiente en Amazon Redshift. A continuación, AWS SCT crea una vista que incluye los datos de todas estas tablas.

  AWS SCT determina automáticamente el número de particiones de la tabla de origen. Según el tipo de particionamiento de la tabla de origen, este número puede superar la cuota de tablas que puede aplicar a su clúster de Amazon Redshift. Para evitar alcanzar esta cuota, introduzca el número máximo de tablas de destino que AWS SCT puede crear para las particiones de una sola

tabla de origen. La opción predeterminada es 368 tablas, lo que representa una partición para 366 días de un año y dos tablas para las particiones `NO RANGE` y `UNKNOWN`.

- Aplicar compresión a las columnas de la tabla de Amazon Redshift. Para ello, seleccione Usar codificación de compresión.

  AWS SCT asigna automáticamente la codificación de compresión a las columnas mediante el algoritmo Amazon Redshift predeterminado. Para obtener más información, consulte Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  De forma predeterminada, Amazon Redshift no aplica compresión a las columnas definidas como claves de clasificación y distribución. Puede cambiar este comportamiento y aplicar compresión a estas columnas. Para ello, seleccione Usar codificación de compresión para columnas KEY. Puede seleccionar esta opción solo si selecciona la opción Usar codificación de compresión.

Configuración de optimización de la conversión de SQL Server Data Warehouse a Amazon Redshift

Para editar la configuración de optimización de la conversión de SQL Server Data Warehouse a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Microsoft SQL Server y, a continuación, Microsoft SQL Server — Amazon Redshift. En el panel izquierdo, elija Estrategias de optimización. AWS SCT muestra la configuración de optimización de la conversión para la conversión de SDL Server Data Warehouse a Amazon Redshift.

La configuración de optimización de la conversión de SQL Server Data Warehouse a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Trabajar con optimización automática de tablas Para ello, seleccione Utilizar el ajuste automático de tablas de Amazon Redshift.

  La optimización automática de tablas es una capacidad de autoajuste presente en Amazon Redshift que optimiza automáticamente el diseño de tablas. Para obtener más información, consulte Trabajar con optimización automática de tablas en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  Para confiar únicamente en la optimización automática de las tablas, elija Ninguna en Estrategia inicial de selección de claves.

- Elegir las claves de clasificación y distribución según su estrategia.

  Puede elegir las claves de clasificación y distribución utilizando metadatos de Amazon Redshift, información estadística o ambas opciones. En estrategia inicial de selección de claves, en la pestaña Estrategias de optimización, elija una de las siguientes opciones:

  - Utilizar metadatos e ignorar la información estadística
  - Ignorar los metadatos, utilizar información estadística
  - Utilizar metadatos e información estadística

  En función de la opción que elija, puede seleccionar estrategias de optimización. A continuación, introduzca el valor (0—100) para cada estrategia. Estos valores definen la ponderación de cada estrategia. Con estos valores de ponderación, AWS SCT define cómo influye cada regla en la elección de las claves de distribución y clasificación. Los valores predeterminados se basan en las prácticas recomendadas de migración de AWS.

  Puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para definirla como una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

- Configurar los detalles de la estrategia.

  Además de definir la ponderación de cada estrategia de optimización, puede configurar los ajustes de optimización. Para ello, elija Optimización de la conversión.

  - En Límite de columnas de clave de clasificación, introduzca el número máximo de columnas en la clave de clasificación.
  - En Valor de umbral descompensado, introduzca el porcentaje (0—100) de un valor descompensado para una columna. AWS SCT excluye de la lista de candidatos a la clave de distribución las columnas con un valor de sesgo superior al umbral. AWS SCT define el valor descompensado de una columna como la relación porcentual entre el número de apariciones del valor más común y el número total de registros.
  - En Consultas N principales de la tabla del historial de consultas, introduzca el número (1–100) de las consultas que se van a analizar con más frecuencia.
  - En Seleccionar usuario de estadísticas, elija el usuario de la base de datos cuyas estadísticas de consulta desee analizar.

Además, en la pestaña Estrategias de optimización, puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para considerarla una tabla pequeña. AWS SCT aplica el estilo de distribución ALL a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

Utilizar Teradata como origen para AWS SCT

Puede utilizar AWS SCT para convertir esquemas, objetos de código y código de aplicaciones de Teradata a Amazon Redshift o Amazon Redshift y AWS Glue utilizados en combinación.

Privilegios para Teradata como origen

Los privilegios necesarios para Teradata como origen se enumeran a continuación:

- SELECT ON DBC
- SELECT ON SYSUDTLIB
- SELECT ON SYSLIB
- SELECT ON *<source_database>*
- CREATE PROCEDURE ON *<source_database>*

En el ejemplo anterior, sustituya el marcador de posición *<source_database>* por el nombre de la base de datos de origen.

AWS SCT requiere el privilegio CREATE PROCEDURE para ejecutar HELP PROCEDURE en todos los procedimientos de la base de datos de origen. AWS SCT no utiliza este privilegio para crear ningún objeto nuevo en la base de datos Teradata de origen.

Conectar a Teradata como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de origen de Teradata con AWS Schema Conversion Tool.

Para conectarse a su base de datos Teradata de origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Elija Teradata y, a continuación, elija Siguiente.

Se abrirá el cuadro de diálogo Agregar origen.

3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.

   - Para introducir manualmente la información de conexión a la base de datos fuente de Teradata, siga las instrucciones siguientes:

Parámetro	Acción
Connection name	Introduzca un nombre para la base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
Database	Escriba el nombre de la base de datos de Teradata.
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseñ

Parámetro	Acción
	a de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Store password	AWS SCT crea un almacén seguro para almacenar certifica dos SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.
Encrypt data	Seleccione esta opción para cifrar los datos que intercamb ie con la base de datos. Si elige esta opción, el número de puerto 443 se utiliza para transferir datos cifrados entre AWS SCT y su base de datos de Teradata.
Teradata driver path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Utilizar la autenticación LDAP con un origen Teradata

Para configurar la autenticación LDAP (Lightweight Directory Access Protocol) para los usuarios de Teradata que ejecutan Microsoft Active Directory en Windows, use el siguiente procedimiento.

En el siguiente procedimiento, el dominio de Active Directory es `test.local.com`. El servidor de Windows es DC y está configurado con la configuración predeterminada. El script de usuario creada en la cuenta de Active Directory `test_ldap` y la cuenta utiliza la contraseña `test_ldap`.

Para configurar la autenticación LDAP para los usuarios de Teradata que ejecutan Microsoft Active Directory en Windows

1. En el directorio `/opt/teradata/tdat/tdgss/site`, edite el archivo `TdgssUserConfigFile.xml`. Cambie la sección LDAP a lo siguiente.

```
AuthorizationSupported="no"

LdapServerName="DC.test.local.com"
LdapServerPort="389"
LdapServerRealm="test.local.com"
LdapSystemFQDN="dc= test, dc= local, dc=com"
LdapBaseFQDN="dc=test, dc=local, dc=com"
```

2. Aplique los cambios ejecutando la configuración como se indica a continuación.

```
#cd /opt/teradata/tdgss/bin
#./run_tdgssconfig
```

3. Pruebe la configuración mediante el siguiente comando.

```
# /opt/teradata/tdat/tdgss/14.10.03.01/bin/tdsbind -u test_ldap -w test_ldap
```

   El resultado debería ser similar al siguiente.

```
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. Reinicie TPA utilizando el siguiente comando.

```
#tpareset -f "use updated TDGSSCONFIG GDO"
```

5. Cree el mismo usuario en la base de datos de Teradata que en Active Directory, tal y como se muestra a continuación.

```
CREATE USER test_ldap AS PERM=1000, PASSWORD=test_ldap;
GRANT LOGON ON ALL TO test WITH NULL PASSWORD;
```

Si cambia la contraseña de usuario en Active Directory para su usuario LDAP, especifique esta nueva contraseña durante la conexión a Teradata en modo LDAP. En el modo DEFAULT, para conectarse a Teradata se utilizan el nombre de usuario y una contraseña de LDAP.

Configurar la recopilación de estadísticas en el almacenamiento de datos de Teradata de origen

Para convertir el almacenamiento de datos de Teradata de origen, AWS SCT utiliza estadísticas para optimizar el almacenamiento de datos convertido de Amazon Redshift. Puede recopilar estadísticas en AWS SCT o cargar el archivo de estadísticas. Para obtener más información, consulte Recopilar o cargar estadísticas.

Para asegurarse de que AWS SCT puede recopilar estadísticas del almacenamiento de datos, complete las siguientes tareas previas.

Para recopilar estadísticas del almacenamiento de datos de Teradata

1. Ejecute la siguiente consulta para recopilar las estadísticas de todas las tablas del almacenamiento de datos.

```
collect summary statistics on table_name;
```

En el ejemplo anterior, sustituya *table_name* por el nombre de la tabla de origen. Repita la consulta para cada tabla que convierta.

2. Ejecute la siguiente consulta para determinar la cadena de cuenta del usuario, que utilizará para convertir el almacenamiento de datos.

```
select * from dbc.accountinfo where username ='user_name'
```

3. Active el registro de consultas para un usuario específico mediante la cadena de cuenta del ejemplo anterior.

```
BEGIN QUERY LOGGING WITH OBJECTS, SQL ON ALL ACCOUNT=('$M$BUSI$S$D$H');
```

También puede activar el registro de consultas para todos los usuarios de la base de datos.

```
BEGIN QUERY LOGGING WITH SQL, OBJECTS LIMIT SQLTEXT=0 ON ALL;
```

Cuando termine de recopilar las estadísticas del almacenamiento de datos, desactive el registro de consultas. Para ello, puede usar el siguiente ejemplo de código.

```
end query logging with explain, objects, sql on all account=(' $M$BUSI$S$D$H');
```

Recopilar estadísticas en modo offline desde su almacenamiento de datos de Teradata de origen

Después de configurar la recopilación de estadísticas en su almacenamiento de datos de Teradata, podrá recopilar estadísticas en su proyecto de AWS SCT. Como alternativa, puede utilizar los scripts Basic Teradata Query (BTEQ) para recopilar estadísticas en modo offline. A continuación, puede cargar los archivos con las estadísticas recopiladas en su proyecto de AWS SCT. Para obtener más información, consulte Recopilar o cargar estadísticas.

Para recopilar estadísticas en modo offline desde el almacenamiento de datos de Teradata de origen

1. Cree el script `off-line_stats.bteq` con el siguiente contenido.

```
.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','BO')
ORDER BY c.TABLENAME, c.COLUMNNAME;

.EXPORT RESET

.EXPORT REPORT FILE = column-skew-stats-tera.csv
```

```
.SET TITLEDASHES OFF
.SET WIDTH 10000

.RUN FILE = column-skew-script-tera.csv

.EXPORT RESET

.EXPORT REPORT FILE = query-stats-tera.csv
.SET TITLEDASHES OFF
.SET WIDTH 32000

SELECT
  '"' || RTRIM(CAST(SqlTextInfo AS VARCHAR(31900)), ';') || '";"' ||
  TRIM(QueryCount) || '";"' ||
  TRIM(QueryId) || '";"' ||
  TRIM(SqlRowNo) || '";"' ||
  TRIM(QueryParts) || '";"' ||
  CAST(CURRENT_TIMESTAMP(0) AS VARCHAR(19)) || '"'
(TITLE
 '"query_text";"query_count";"query_id";"sql_row_no";"query_parts";"current_ts"')
  FROM
  (
    SELECT  QueryId,  SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
    SUM(QueryFirstRow) OVER (ORDER BY QueryCount DESC, QueryId ASC, SqlRowNo ASC
    ROWS UNBOUNDED PRECEDING) AS topN
    FROM
    (SELECT QueryId,  SqlTextInfo, SqlRowNo, QueryParts, QueryCount,
      CASE WHEN
      ROW_NUMBER() OVER (PARTITION BY QueryCount, SqlTextInfo ORDER BY QueryId,
 SqlRowNo) = 1 AND SqlRowNo = 1
    THEN 1 ELSE 0 END AS QueryFirstRow
    FROM (
      SELECT q.QueryId,  q.SqlTextInfo, q.SqlRowNo,
      MAX(q.SqlRowNo) OVER (PARTITION BY q.QueryId) QueryParts,
      COUNT(q.SqlTextInfo) OVER (PARTITION BY q.SqlTextInfo) QueryCount
      FROM DBC.dbqlsqltbl q
      INNER JOIN
      (
        SELECT QueryId
        FROM DBC.DBQLogTbl t
        WHERE TRIM(t.StatementType) IN ('SELECT')
        AND TRIM(t.AbortFlag) = '' AND t.ERRORCODE = 0
        AND  (CASE WHEN 'All users' IN ('All users') THEN 'All users' ELSE
 TRIM(t.USERNAME) END) IN ('All users') --user_name list
```

```
        AND t.StartTime > CURRENT_TIMESTAMP - INTERVAL '30' DAY
        GROUP BY 1
      ) t
      ON q.QueryId = t.QueryId
      INNER JOIN
      (
        SELECT QueryId
        FROM DBC.QryLogObjectsV
        WHERE ObjectDatabaseName = 'your_database_name'
        AND ObjectType = 'Tab'
        AND CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
        GROUP BY 1
      ) r
      ON r.QueryId = t.QueryId
      WHERE q.CollectTimeStamp > CURRENT_TIMESTAMP - INTERVAL '30' DAY
    ) t
  ) t
  WHERE SqlTextInfo NOT LIKE '%";"%'
  ) q
  WHERE
  QueryParts >=1
  AND topN <= 50
  ORDER BY QueryCount DESC, QueryId, SqlRowNo
  QUALIFY COUNT(QueryId) OVER (PARTITION BY QueryId) = QueryParts;

.EXPORT RESET

.LOGOFF

.QUIT
```

2. Cree el archivo `td_run_bteq.bat` que ejecuta el script de BTEQ que creó en el paso anterior. Utilice el siguiente contenido para este archivo.

```
@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. Cree el archivo `runme.bat` que ejecuta el archivo de procesamiento por lotes que creó en el paso anterior. Utilice el siguiente contenido para este archivo.

```
.\td_run_bteq.bat ServerName UserName DatabaseName
```

En el archivo `runme.bat`, sustituya *`Servername`*, UserName y *`DatabaseName`* por los valores aplicables.

A continuación, ejecute el archivo de `runme.bat`. Repita este paso para cada almacenamiento de datos que convierta en Amazon Redshift.

Tras ejecutar este script, recibirá tres archivos con estadísticas para cada base de datos. Puede cargar estos archivos en su proyecto de AWS SCT. Para ello, elija el almacenamiento de datos en el panel izquierdo del proyecto y, a continuación, abra el menú contextual (clic secundario). Seleccione Cargar estadísticas.

Configuración de conversión de Teradata a Amazon Redshift

Para editar la configuración de conversión de Teradata a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Teradata y, a continuación, elija Teradata — Amazon Redshift. AWS SCT muestra todos los ajustes disponibles para la conversión de Teradata a Amazon Redshift.

La configuración de conversión de Teradata a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.
- Establecer el número máximo de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift de destino.

  En El número máximo de tablas para el clúster de Amazon Redshift de destino, elija el número de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift.

Amazon Redshift tiene cuotas que limitan las tablas de uso para los distintos tipos de nodos de clúster. Si elige Auto, AWS SCT determina el número de tablas que se van a aplicar al clúster de Amazon Redshift de destino en función del tipo de nodo. Si lo desea, elija el valor manualmente. Para obtener más información, consulte Cuotas y límites de Amazon Redshift en la Guía de administración de Amazon Redshift.

AWS SCT convierte todas las tablas de origen, aunque sean más de las que su clúster de Amazon Redshift puede almacenar. AWS SCTalmacena el código convertido en su proyecto y no lo aplica a la base de datos de destino. Si alcanza la cuota del clúster de Amazon Redshift para las tablas al aplicar el código convertido, AWS SCT mostrará un mensaje de advertencia. Además, AWS SCT aplica tablas a su clúster de Amazon Redshift de destino hasta que el número de tablas alcanza el límite.

- Migrar particiones de la tabla de origen a tablas independientes en Amazon Redshift. Para ello, seleccione Usar la vista UNION ALL e introduzca el número máximo de tablas de destino que AWS SCT puede crear para una sola tabla de origen.

  Amazon Redshift no admite particionamiento de tablas. Para simular este comportamiento y agilizar la ejecución de las consultas, AWS SCT puede migrar cada partición de la tabla de origen a una tabla independiente en Amazon Redshift. A continuación, AWS SCT crea una vista que incluye los datos de todas estas tablas.

  AWS SCT determina automáticamente el número de particiones de la tabla de origen. Según el tipo de particionamiento de la tabla de origen, este número puede superar la cuota de tablas que puede aplicar a su clúster de Amazon Redshift. Para evitar alcanzar esta cuota, introduzca el número máximo de tablas de destino que AWS SCT puede crear para las particiones de una sola tabla de origen. La opción predeterminada es 368 tablas, lo que representa una partición para 366 días de un año y dos tablas para las particiones `NO RANGE` y `UNKNOWN`.
- Aplicar compresión a las columnas de la tabla de Amazon Redshift. Para ello, seleccione Usar codificación de compresión.

  AWS SCT asigna automáticamente la codificación de compresión a las columnas mediante el algoritmo Amazon Redshift predeterminado. Para obtener más información, consulte Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  De forma predeterminada, Amazon Redshift no aplica compresión a las columnas definidas como claves de clasificación y distribución. Puede cambiar este comportamiento y aplicar compresión a

estas columnas. Para ello, seleccione Usar codificación de compresión para columnas KEY. Puede seleccionar esta opción solo si selecciona la opción Usar codificación de compresión.

- Para usar una lista explícita de columnas en el código convertido para las instrucciones `SELECT *`, seleccione Usar instrucción de columna explícita.
- Para simular el comportamiento de las claves principales y únicas del clúster de Amazon Redshift, seleccione Simular el comportamiento de claves principales y únicas.

  Amazon Redshift no exige claves únicas y principales y las utiliza únicamente con fines informativos. Si utiliza estas restricciones en el código, asegúrese de que AWS SCT simula su comportamiento en el código convertido.
- Garantizar la exclusividad de los datos en las tablas de Amazon Redshift de destino. Para ello, seleccione Simular el comportamiento de las tablas SET.

  Teradata crea tablas utilizando el elemento de sintaxis `SET` como opción predeterminada. No puede añadir filas duplicadas en una tabla `SET`. Si su código fuente no usa esta restricción de exclusividad, desactive esta opción. En este caso, el código convertido funciona más rápido.

  Si su código fuente usa la opción `SET` de las tablas como una restricción de exclusividad, active esta opción. En este caso, AWS SCT reescribe las instrucciones `INSERT..SELECT` en el código convertido para simular el comportamiento de la base de datos de origen.

Configuración de optimización de la conversión de Teradata a Amazon Redshift

Para editar la configuración de optimización de la conversión de Teradata a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Teradata y, a continuación, Teradata — Amazon Redshift. En el panel izquierdo, elija Estrategias de optimización. AWS SCT muestra la configuración de optimización de la conversión para la conversión de Teradata a Amazon Redshift.

La configuración de optimización de la conversión de Teradata a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Trabajar con optimización automática de tablas Para ello, seleccione Utilizar el ajuste automático de tablas de Amazon Redshift.

  La optimización automática de tablas es una capacidad de autoajuste presente en Amazon Redshift que optimiza automáticamente el diseño de tablas. Para obtener más información,

consulte Trabajar con optimización automática de tablas en la Guía para desarrolladores de bases de datos de Amazon Redshift.

Para confiar únicamente en la optimización automática de las tablas, elija Ninguna en Estrategia inicial de selección de claves.

- Elegir las claves de clasificación y distribución según su estrategia.

  Puede elegir las claves de clasificación y distribución utilizando metadatos de Amazon Redshift, información estadística o ambas opciones. En estrategia inicial de selección de claves, en la pestaña Estrategias de optimización, elija una de las siguientes opciones:

  - Utilizar metadatos e ignorar la información estadística
  - Ignorar los metadatos, utilizar información estadística
  - Utilizar metadatos e información estadística

  En función de la opción que elija, puede seleccionar estrategias de optimización. A continuación, introduzca el valor (0—100) para cada estrategia. Estos valores definen la ponderación de cada estrategia. Con estos valores de ponderación, AWS SCT define cómo influye cada regla en la elección de las claves de distribución y clasificación. Los valores predeterminados se basan en las prácticas recomendadas de migración de AWS.

  Puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para definirla como una tabla pequeña. AWS SCT aplica el estilo de distribución ALL a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

- Configurar los detalles de la estrategia.

  Además de definir la ponderación de cada estrategia de optimización, puede configurar los ajustes de optimización. Para ello, elija Optimización de la conversión.

  - En Límite de columnas de clave de clasificación, introduzca el número máximo de columnas en la clave de clasificación.
  - En Valor de umbral descompensado, introduzca el porcentaje (0—100) de un valor descompensado para una columna. AWS SCT excluye de la lista de candidatos a la clave de distribución las columnas con un valor de sesgo superior al umbral. AWS SCT define el valor descompensado de una columna como la relación porcentual entre el número de apariciones del valor más común y el número total de registros.

- En Consultas N principales de la tabla del historial de consultas, introduzca el número (1–100) de las consultas que se van a analizar con más frecuencia.
- En Seleccionar usuario de estadísticas, elija el usuario de la base de datos cuyas estadísticas de consulta desee analizar.

Además, en la pestaña Estrategias de optimización, puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para considerarla una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

Utilizar Vertica como origen para AWS SCT

Puede utilizar AWS SCT para convertir esquemas, objetos de código y código de aplicación de Vertica a Amazon Redshift.

Privilegios para Vertica como origen

Los privilegios necesarios para Vertica como origen se enumeran a continuación:

- USAGE ON SCHEMA *<schema_name>*
- USAGE ON SCHEMA PUBLIC
- SELECT ON ALL TABLES IN SCHEMA *<schema_name>*
- SELECT ON ALL SEQUENCES IN SCHEMA *<schema_name>*
- EXECUTE ON ALL FUNCTIONS IN SCHEMA *<schema_name>*
- EXECUTE ON PROCEDURE *<schema_name.procedure_name(procedure_signature)>*

En el ejemplo anterior, sustituya los marcadores de posición del modo que sigue:

- Sustituya *schema_name* por el nombre del esquema de origen.
- Sustituya *procedure_name* por el nombre de un procedimiento de origen. Repita la concesión para cada procedimiento que vaya a convertir.
- Sustituya *procedure_signature* por la lista de tipos de argumentos de procedimiento delimitada por comas.

Conectar a Vertica como origen

Utilice el siguiente procedimiento para conectarse a su base de datos de origen de Vertica con AWS Schema Conversion Tool.

Para conectarse a su base de datos Vertica de origen

1. En AWS Schema Conversion Tool, seleccione Agregar origen.
2. Seleccione Vertica y, a continuación, seleccione Siguiente.

   Se abrirá el cuadro de diálogo Agregar origen.
3. En Nombre de conexión, escriba un nombre para su base de datos. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. Utilice las credenciales de la base de datos de AWS Secrets Manager o introdúzcalas manualmente:

   - Para usar las credenciales de base de datos de Secrets Manager, siga las instrucciones siguientes:

     1. En Secreto de AWS, elija el nombre del secreto.
     2. Seleccione Rellenar para rellenar automáticamente todos los valores del cuadro de diálogo de conexión a la base de datos de Secrets Manager.

     Para obtener información sobre el uso de las credenciales de bases de datos de Secrets Manager, consulteUtilizar AWS Secrets Manager.
   - Para introducir manualmente la información de conexión a la base de datos fuente de Vertica, siga las instrucciones siguientes:

Parámetro	Acción
Server name	Escriba el nombre del sistema de nombres de dominio (DNS) o la dirección IP del servidor de base de datos de origen.
Server port	Escriba el puerto utilizado para conectarse al servidor de base de datos de origen.
Database	Escriba el nombre de la base de datos de Vertica.

Parámetro	Acción
User name y Password	Introduzca las credenciales de la base de datos para conectarse al servidor de base de datos de origen. AWS SCT utiliza la contraseña para conectarse a la base de datos de origen solo cuando decide conectarse a la base de datos en un proyecto. Para evitar que la contraseña de la base de datos de origen quede expuesta, AWS SCT no la almacena por defecto. Si cierra su proyecto de AWS SCT y vuelve a abrirlo, se le solicitará la contraseña para conectarse a su base de datos de origen, según sea necesario.
Use SSL	Seleccione esta opción para utilizar capa de sockets seguros (SSL) para conectarse a la base de datos. Proporcione la siguiente información adicional, según proceda, en la pestaña SSL: • Verify server certificate: seleccione esta opción para verificar el certificado del servidor mediante un almacén de confianza. • Trust store: un almacén de confianza que haya configurado en Configuración global. • Key store: un almacén de claves que haya configurado en Configuración global.
Store password	AWS SCT crea un almacén seguro para almacenar certificados SSL y contraseñas de la base de datos. Puede habilitar esta opción para almacenar la contraseña de la base de datos y conectarse rápidamente a la base de datos sin necesidad de introducir la contraseña.

Parámetro	Acción
Vertica driver path	Escriba la ruta al controlador que va a usar para conectars e a la base de datos de origen. Para obtener más informaci ón, consulte Descarga de los controladores de base de datos necesarios. Si almacena la ruta al controlador en la configuración global del proyecto, la ruta del controlador no aparecerá en el cuadro de diálogo de conexión. Para obtener más informaci ón, consulte Almacenamiento de rutas a los controladores en la configuración global.

5. Seleccione Probar la conexión para verificar que AWS SCT pueda conectarse a su base de datos de origen.
6. Seleccione Conectar para conectarse a su base de datos de origen.

Configuración de conversión de Vertica a Amazon Redshift

Para editar la configuración de conversión de Vertica a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Vertica y, a continuación, elija Vertica — Amazon Redshift. AWS SCT muestra todos los ajustes disponibles para la conversión de Vertica a Amazon Redshift.

La configuración de conversión de Vertica a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Limitar el número de comentarios con elementos de acción en el código convertido.

  En Añadir comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior, seleccione la gravedad de los elementos de acción. AWS SCTañade comentarios en el código convertido para los elementos de acción de la gravedad seleccionada o superior.

  Por ejemplo, para minimizar el número de comentarios en el código convertido, seleccione Solo errores. Para incluir comentarios para todos los elementos de acción del código convertido, seleccione Todos los mensajes.

- Establecer el número máximo de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift de destino.

  En El número máximo de tablas para el clúster de Amazon Redshift de destino, elija el número de tablas que AWS SCT puede aplicar a su clúster de Amazon Redshift.

  Amazon Redshift tiene cuotas que limitan las tablas de uso para los distintos tipos de nodos de clúster. Si elige Auto, AWS SCT determina el número de tablas que se van a aplicar al clúster de Amazon Redshift de destino en función del tipo de nodo. Si lo desea, elija el valor manualmente. Para obtener más información, consulte Cuotas y límites de Amazon Redshift en la Guía de administración de Amazon Redshift.

  AWS SCT convierte todas las tablas de origen, aunque sean más de las que su clúster de Amazon Redshift puede almacenar. AWS SCTalmacena el código convertido en su proyecto y no lo aplica a la base de datos de destino. Si alcanza la cuota del clúster de Amazon Redshift para las tablas al aplicar el código convertido, AWS SCT mostrará un mensaje de advertencia. Además, AWS SCT aplica tablas a su clúster de Amazon Redshift de destino hasta que el número de tablas alcanza el límite.
- Migrar particiones de la tabla de origen a tablas independientes en Amazon Redshift. Para ello, seleccione Usar la vista UNION ALL e introduzca el número máximo de tablas de destino que AWS SCT puede crear para una sola tabla de origen.

  Amazon Redshift no admite particionamiento de tablas. Para simular este comportamiento y agilizar la ejecución de las consultas, AWS SCT puede migrar cada partición de la tabla de origen a una tabla independiente en Amazon Redshift. A continuación, AWS SCT crea una vista que incluye los datos de todas estas tablas.

  AWS SCT determina automáticamente el número de particiones de la tabla de origen. Según el tipo de particionamiento de la tabla de origen, este número puede superar la cuota de tablas que puede aplicar a su clúster de Amazon Redshift. Para evitar alcanzar esta cuota, introduzca el número máximo de tablas de destino que AWS SCT puede crear para las particiones de una sola tabla de origen. La opción predeterminada es 368 tablas, lo que representa una partición para 366 días de un año y dos tablas para las particiones `NO RANGE` y `UNKNOWN`.
- Aplicar compresión a las columnas de la tabla de Amazon Redshift. Para ello, seleccione Usar codificación de compresión.

  AWS SCT asigna automáticamente la codificación de compresión a las columnas mediante el algoritmo Amazon Redshift predeterminado. Para obtener más información, consulte

Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.

De forma predeterminada, Amazon Redshift no aplica compresión a las columnas definidas como claves de clasificación y distribución. Puede cambiar este comportamiento y aplicar compresión a estas columnas. Para ello, seleccione Usar codificación de compresión para columnas KEY. Puede seleccionar esta opción solo si selecciona la opción Usar codificación de compresión.

Configuración de optimización de la conversión de Vertica a Amazon Redshift

Para editar la configuración de optimización de la conversión de Vertica a Amazon Redshift, seleccione Configuración en AWS SCT y, a continuación, elija Configuración de conversión. En la lista superior, elija Vertica y, a continuación, Vertica — Amazon Redshift. En el panel izquierdo, elija Estrategias de optimización. AWS SCT muestra la configuración de optimización de la conversión para la conversión de Vertica a Amazon Redshift.

La configuración de optimización de la conversión de Vertica a Amazon Redshift en AWS SCT incluye opciones para lo siguiente:

- Trabajar con optimización automática de tablas Para ello, seleccione Utilizar el ajuste automático de tablas de Amazon Redshift.

  La optimización automática de tablas es una capacidad de autoajuste presente en Amazon Redshift que optimiza automáticamente el diseño de tablas. Para obtener más información, consulte Trabajar con optimización automática de tablas en la Guía para desarrolladores de bases de datos de Amazon Redshift.

  Para confiar únicamente en la optimización automática de las tablas, elija Ninguna en Estrategia inicial de selección de claves.
- Elegir las claves de clasificación y distribución según su estrategia.

  Puede elegir las claves de clasificación y distribución utilizando metadatos de Amazon Redshift, información estadística o ambas opciones. En estrategia inicial de selección de claves, en la pestaña Estrategias de optimización, elija una de las siguientes opciones:

  - Utilizar metadatos e ignorar la información estadística
  - Ignorar los metadatos, utilizar información estadística
  - Utilizar metadatos e información estadística

En función de la opción que elija, puede seleccionar estrategias de optimización. A continuación, introduzca el valor (0—100) para cada estrategia. Estos valores definen la ponderación de cada estrategia. Con estos valores de ponderación, AWS SCT define cómo influye cada regla en la elección de las claves de distribución y clasificación. Los valores predeterminados se basan en las prácticas recomendadas de migración de AWS.

Puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para definirla como una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

- Configurar los detalles de la estrategia.

  Además de definir la ponderación de cada estrategia de optimización, puede configurar los ajustes de optimización. Para ello, elija Optimización de la conversión.

  - En Límite de columnas de clave de clasificación, introduzca el número máximo de columnas en la clave de clasificación.
  - En Valor de umbral descompensado, introduzca el porcentaje (0—100) de un valor descompensado para una columna. AWS SCT excluye de la lista de candidatos a la clave de distribución las columnas con un valor de sesgo superior al umbral. AWS SCT define el valor descompensado de una columna como la relación porcentual entre el número de apariciones del valor más común y el número total de registros.
  - En Consultas N principales de la tabla del historial de consultas, introduzca el número (1–100) de las consultas que se van a analizar con más frecuencia.
  - En Seleccionar usuario de estadísticas, elija el usuario de la base de datos cuyas estadísticas de consulta desee analizar.

  Además, en la pestaña Estrategias de optimización, puede definir el tamaño de las tablas pequeñas para la estrategia Buscar tablas pequeñas. En Recuento mínimo de filas de la tabla y Recuento máximo de filas de la tabla, introduzca el número mínimo y máximo de filas de una tabla para considerarla una tabla pequeña. AWS SCT aplica el estilo de distribución `ALL` a las tablas pequeñas. En este caso, se distribuye una copia de toda la tabla a cada nodo.

Creación de reglas de asignación en AWS SCT

Puede añadir varias bases de datos de origen y destino en un solo proyecto de AWS SCT. De este modo, se simplifica la administración de los proyectos al migrar varias bases de datos a diferentes plataformas de destino.

Tras crear un proyecto nuevo y añadir las bases de datos de origen y destino, cree reglas de asignación. AWS SCT requiere al menos una regla de asignación para crear un informe de evaluación de la migración y convertir los esquemas de bases de datos.

Una regla de asignación describe un par origen-destino que incluye un esquema de base de datos de origen o una base de datos de origen y una plataforma de base de datos de destino. Puede crear varias reglas de asignación en un solo proyecto de AWS SCT. Utilice las reglas de asignación para convertir cada esquema de base de datos de origen en la plataforma de base de datos de destino correcta.

Para cambiar el nombre del esquema en el código convertido, configure una regla de migración. Por ejemplo, con las reglas de migración, puede cambiar el nombre del esquema, agregar un prefijo a los nombres de los objetos, cambiar la intercalación de columnas o cambiar los tipos de datos. Para aplicar estos cambios al código convertido, cree reglas de migración antes de convertir el esquema de origen. Para obtener más información, consulte Crear reglas de migración.

Puede crear reglas de asignación solo para los pares de conversión de bases de datos compatibles. Para ver la lista de pares de conversión compatibles, consulteOrígenes para AWS SCT.

Si abre un proyecto guardado en la versión 1.0.655 o anterior de AWS SCT, AWS SCT crea automáticamente reglas de asignación para todos los esquemas de la base de datos de origen a la plataforma de base de datos de destino. Para añadir otras plataformas de bases de datos de destino, elimine las reglas de asignación existentes y, a continuación, cree reglas de asignación nuevas.

Temas

- Cómo agregar una regla de asignación nueva
- Administración de reglas de asignación
- Usar destinos virtuales
- Limitaciones al uso de varios servidores en un solo proyecto de AWS SCT

Cómo agregar una regla de asignación nueva

Puede crear varias reglas de asignación en un solo proyecto. AWS SCT guarda reglas de asignación como parte del proyecto. Con el proyecto abierto, utilice el siguiente procedimiento para crear reglas de asignación.

Para crear reglas de asignación

1. En el menú Ver, seleccione Vista principal.
2. En el panel izquierdo, elija un esquema o una base de datos para agregarlos a la regla de asignación.
3. En el panel derecho, elija una plataforma de base de datos de destino para el esquema o la base de datos de origen seleccionados.

   Puede elegir una plataforma de base de datos virtual como destino. Para obtener más información, consulte Usar destinos virtuales.
4. Seleccione Crear asignación.

   AWS SCT agrega esta regla de asignación nueva a la lista Asignaciones de servidor.

Agregue reglas de asignación para todos los pares de conversión. Para crear un informe de evaluación o convertir esquemas de bases de datos, seleccione Vista principal en el menú Ver.

AWS SCT resalta en negrita todos los objetos del esquema que forman parte de una regla de asignación.

Administración de reglas de asignación

Puede filtrar o eliminar las reglas de asignación existentes y agregar una regla de asignación nueva al proyecto de AWS Schema Conversion Tool (AWS SCT).

Al crear una regla de asignación para toda la base de datos de origen, AWS SCT crea una regla de asignación para cada esquema de base de datos de origen. En el caso de proyectos que incluyen docenas de esquemas o incluso bases de datos, puede resultar difícil entender qué destino se utiliza para un esquema determinado. Para encontrar rápidamente una regla de asignación para el esquema, utilice una o varias de las siguientes opciones de filtro en AWS SCT.

Para filtrar reglas de asignación

1. En el menú Ver, seleccione Vista principal.
2. En Base de datos de origen, seleccione su base de datos de origen.

   El filtro predeterminado es Todos, lo que significa que AWS SCT muestra las reglas de asignación para todas las bases de datos de origen.
3. En Esquema de origen, introduzca el nombre del esquema de origen. Utilice el porcentaje (%) como carácter comodín para reemplazar los símbolos del nombre del esquema.

   El filtro predeterminado es %, lo que significa que AWS SCT muestra las reglas de asignación para todos los nombres de esquemas de bases de datos de origen.
4. En Tiene reglas de migración, seleccione Sí para mostrar las reglas de asignación para las que se han creado reglas de migración de datos. Elija No para mostrar las reglas de asignación que no tienen reglas de migración de datos. Para obtener más información, consulte Crear reglas de migración de datos en AWS SCT.

   El filtro predeterminado es Todos, lo que significa que AWS SCT muestra todas las reglas de asignación.
5. En Servidores de destino, elija la base de datos de destino.

   El filtro predeterminado es Todos, lo que significa que AWS SCT muestra las reglas de asignación para todas las bases de datos de destino.

Con el proyecto abierto, utilice el siguiente procedimiento para eliminar una regla de asignación. Para obtener más información sobre las reglas de asignación, consulte Cómo agregar una regla de asignación nueva.

Para eliminar reglas de asignación

1. En el menú Ver, seleccione Vista principal.
2. En Asignaciones de servidor, elija las reglas de asignación que desee eliminar.
3. Elija Eliminar asignaciones seleccionadas.

   AWS SCT elimina las reglas de asignación seleccionadas.

Usar destinos virtuales

Puede ver cómo AWS SCT convierte el esquema de su base de datos de origen a cualquier plataforma de base de datos de destino compatible. Para hacerlo, no necesita conectarse a una base de datos de destino existente. En su lugar, puede elegir una plataforma de base de datos de destino virtual en el panel derecho al crear una regla de asignación. Para obtener más información, consulte Cómo agregar una regla de asignación nueva. Expanda los nodos Servidores, Clústeres de NoSQL y ETL en el panel derecho para ver la lista de plataformas de bases de datos de destino virtuales.

AWS SCT admite las siguientes plataformas de bases de datos de destino virtuales:

- Amazon Aurora MySQL-Compatible Edition
- Amazon Aurora PostgreSQL-Compatible Edition
- Amazon DynamoDB
- Amazon Redshift
- Amazon Redshift y AWS Glue
- AWS Glue
- AWS Glue Studio
- Babelfish for Aurora PostgreSQL
- MariaDB
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL

Si utiliza Babelfish para Aurora PostgreSQL como plataforma de base de datos de destino, solo puede crear un informe de evaluación de la migración de la base de datos. Para obtener más información, consulte the section called “Informes de evaluación de migración”.

Si utiliza una plataforma de base de datos de destino virtual, puede guardar el código convertido en un archivo. Para obtener más información, consulte the section called “Cómo guardar el esquema convertido”.

Limitaciones al uso de varios servidores en un solo proyecto de AWS SCT

Las siguientes limitaciones se aplican al convertir esquemas usando varios servidores en un solo proyecto de AWS SCT:

- Puede añadir el mismo servidor a un proyecto solo una vez.
- No puede asignar los esquemas del servidor a un esquema de destino específico, solo a un servidor de destino. AWS SCT crea el esquema de destino durante la conversión.
- No puede asignar objetos de origen de nivel inferior al servidor de destino.
- Puede asignar un esquema de origen a un solo servidor de destino de un proyecto.
- Asigne una origen a un servidor de destino para crear un informe de evaluación, convertir esquemas o extraer datos.

Creación de informes de conversión

Cuando se programa una conversión de base de datos, resulta útil crear algunos informes que ayuden a entender todo lo que implica. Puede crear informes usando AWS Schema Conversion Tool.

Puede utilizar AWS SCT para crear un informe de evaluación de la migración de la base de datos. Con este informe, obtendrá un resumen de sus tareas de conversión del esquema y detalles de elementos que no se pueden convertir automáticamente a la base de datos de destino. Puede utilizar este informe para evaluar qué parte del proyecto se puede realizar con AWS SCT, y qué más tiene que hacer para completar la conversión. Para crear un informe de evaluación, utilice Create Report (Crear informe) en el menú contextual (clic con el botón derecho) de la base de datos en AWS SCT.

Temas

- Crear informes de evaluación de migración con AWS SCT

Crear informes de evaluación de migración con AWS SCT

Una parte importante de AWS Schema Conversion Tool es el informe de evaluación que genera para estimar la complejidad de la conversión del esquema. Este informe de evaluación de migración de base de datos resume todas las tareas de conversión del esquema y detalla los elementos de acción del esquema que no se hayan podido convertir al motor de la base de datos de su instancia de destino. Puede ver el informe en la aplicación o exportarlo como un archivo de valores separados por comas (CSV) o PDF.

Si agrega varias bases de datos de origen y destino en un solo proyecto, AWS SCT agrega los informes de todos los pares de conversiones en un solo informe de evaluación de la migración de la base de datos.

Puede utilizar plataformas de bases de datos de destino virtuales para generar un informe de evaluación y comprender la complejidad de la migración a una plataforma de base de datos seleccionada. En este caso, no es necesario conectarse a la plataforma de base de datos de destino. Por ejemplo, puede utilizar Babelfish para Aurora PostgreSQL como plataforma de base de datos de destino virtual para crear un informe de evaluación de migración de la base de datos. Para obtener más información sobre las plataformas de bases de datos de destino virtuales, consulte the section called “Destinos virtuales”.

El informe de evaluación de la migración incluye lo siguiente:

- Resumen ejecutivo
- Evaluación de licencias
- Compatibilidad con la nube, que indica las características de la base de datos de origen no disponibles en el destino
- Recomendaciones, incluida la conversión de objetos de servidor, sugerencias de backup y cambios en el servidor enlazado

El informe incluye también estimaciones de la cantidad de esfuerzo necesario para escribir el código equivalente para su instancia de base de datos de destino que no se pudo convertir automáticamente.

Si utiliza AWS SCT para migrar su esquema existente a una instancia de base de datos de Amazon RDS, puede usar el informe para analizar los requisitos para migrar a la nube de AWS y para cambiar el tipo de licencia.

Temas

- Creación de un informe de evaluación de la migración de la base de datos
- Ver el informe de evaluación
- Guardar el informe de evaluación
- Configurar el informe de evaluación
- Crear un informe de evaluación multiservidor para la migración de la base de datos

Creación de un informe de evaluación de la migración de la base de datos

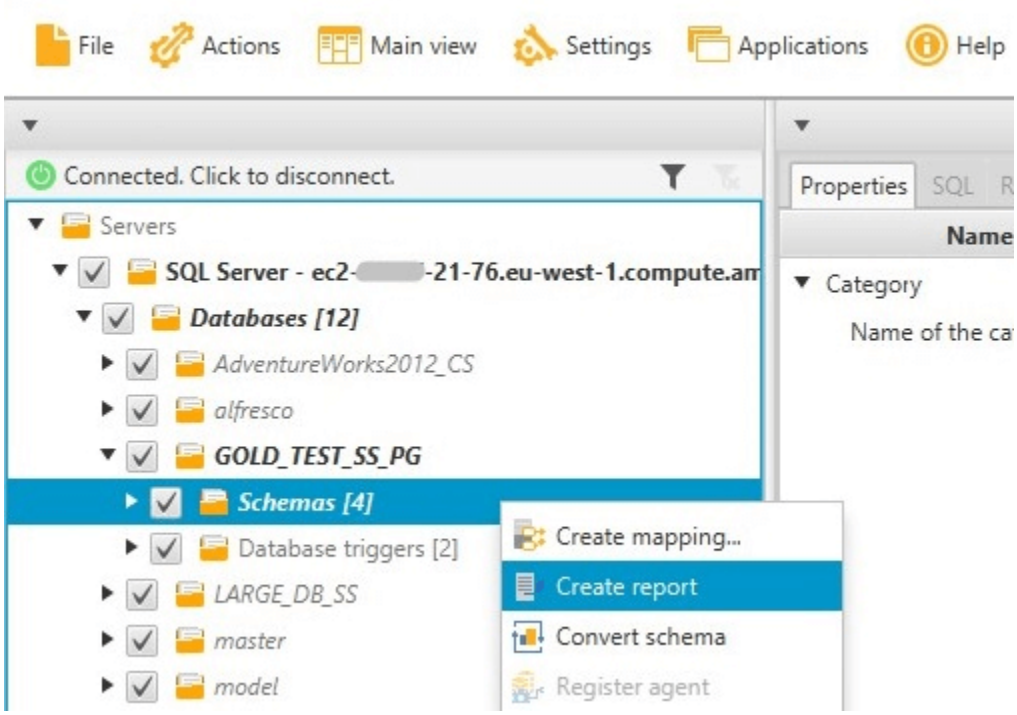
Utilice el siguiente procedimiento para crear un informe de evaluación de la migración de la base de datos.

Para crear un informe de evaluación de la migración de la base de datos

1. Asegúrese de haber creado una regla de asignación para el esquema de la base de datos de origen para crear un informe de evaluación. Para obtener más información, consulte Cómo agregar una regla de asignación nueva.
2. En el menú Ver, seleccione Vista principal.

3. En el panel de la izquierda que muestra el esquema de la base de datos de origen, elija el objeto del esquema para el que desea crear un informe de evaluación. Para incluir varios esquemas de bases de datos en el informe, elija el nodo principal, por ejemplo, Esquemas.

   Asegúrese de haber seleccionado las casillas de verificación de todos los objetos del esquema para los que desee crear un informe de evaluación.
4. Abra el menú contextual (clic secundario) para el objeto y seleccione Crear informe.



Ver el informe de evaluación

Tras crear un informe de evaluación, se abrirá la vista del informe de evaluación, mostrando las siguientes pestañas.

- Resumen
- Elementos de acción

En la pestaña Resumen se muestran los elementos convertidos automáticamente o no convertidos.

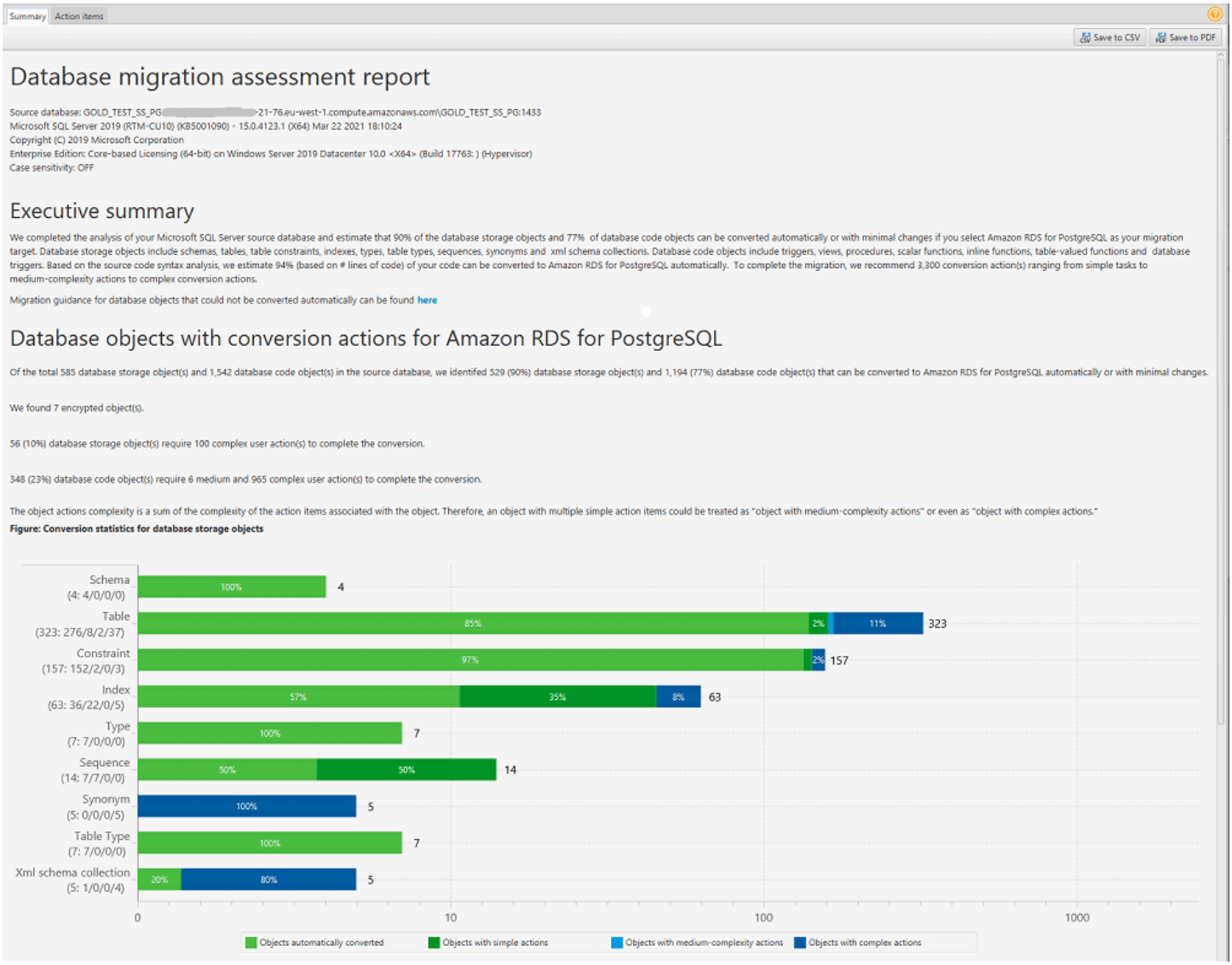
La pestaña Elementos de acción muestra los elementos que no se han podido convertir automáticamente y recomendaciones sobre qué hacer con ellos.

Temas

- Resumen de un informe de evaluación
- Elementos de acción del informe de evaluación
- Mensaje de advertencia del informe de evaluación

Resumen de un informe de evaluación

En la pestaña Resumen se muestra la información resumida del informe de evaluación de la migración de la base de datos. Muestra los elementos que se hayan convertido automáticamente y los elementos que no se hayan convertido automáticamente.



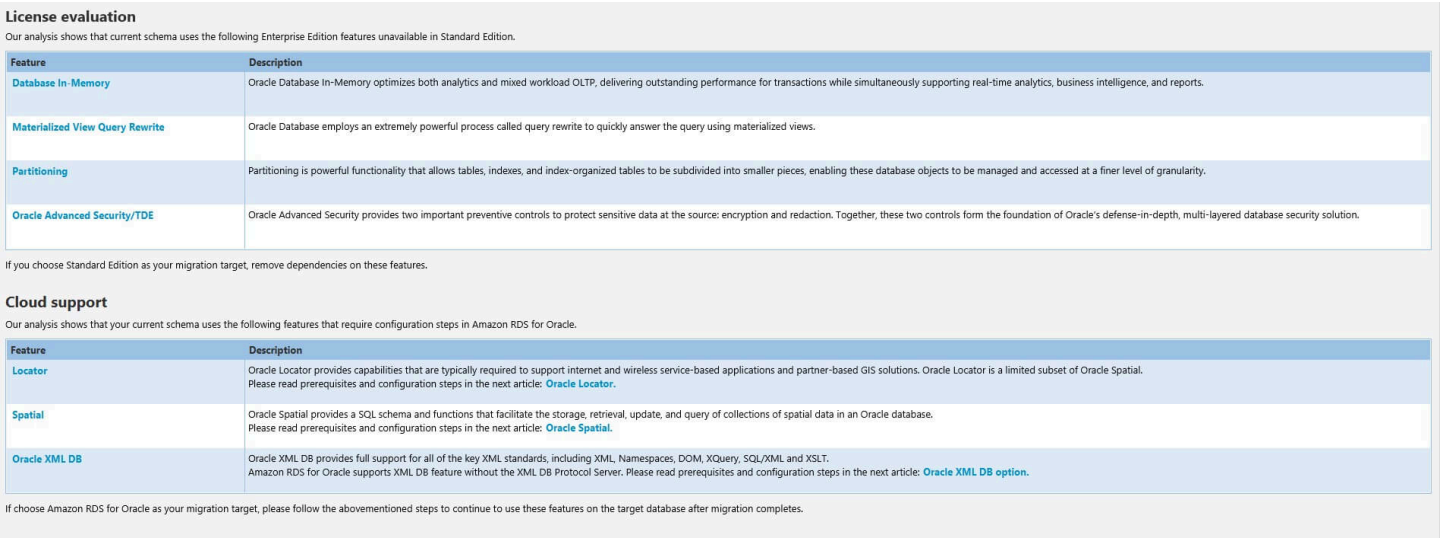
Para los elementos del esquema que no se puedan convertir automáticamente al motor de base de datos de destino, el resumen incluye una estimación del esfuerzo necesario para crear elementos

del esquema en su instancia de base de datos de destino que sean equivalentes a los de su base de datos de origen.

En el informe, el tiempo estimado para convertir estos elementos del esquema se clasifica en los siguientes grupos:

- Simple: acciones que se pueden realizar en menos de una hora.
- Intermedio: acciones que son más complejas y se pueden realizar en el plazo de una a cuatro horas.
- Significativo: acciones que son muy complejas y requieren más de cuatro horas.

La sección Evaluación de licencias y soporte en la nube contiene información sobre el traslado de sus esquemas de bases de datos en las instalaciones existentes a una instancia de base de datos de Amazon RDS DB que ejecute el mismo motor. Por ejemplo, si desea cambiar tipos de licencia, esta sección del informe le indica qué características de su base de datos actual deberían eliminarse.



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.

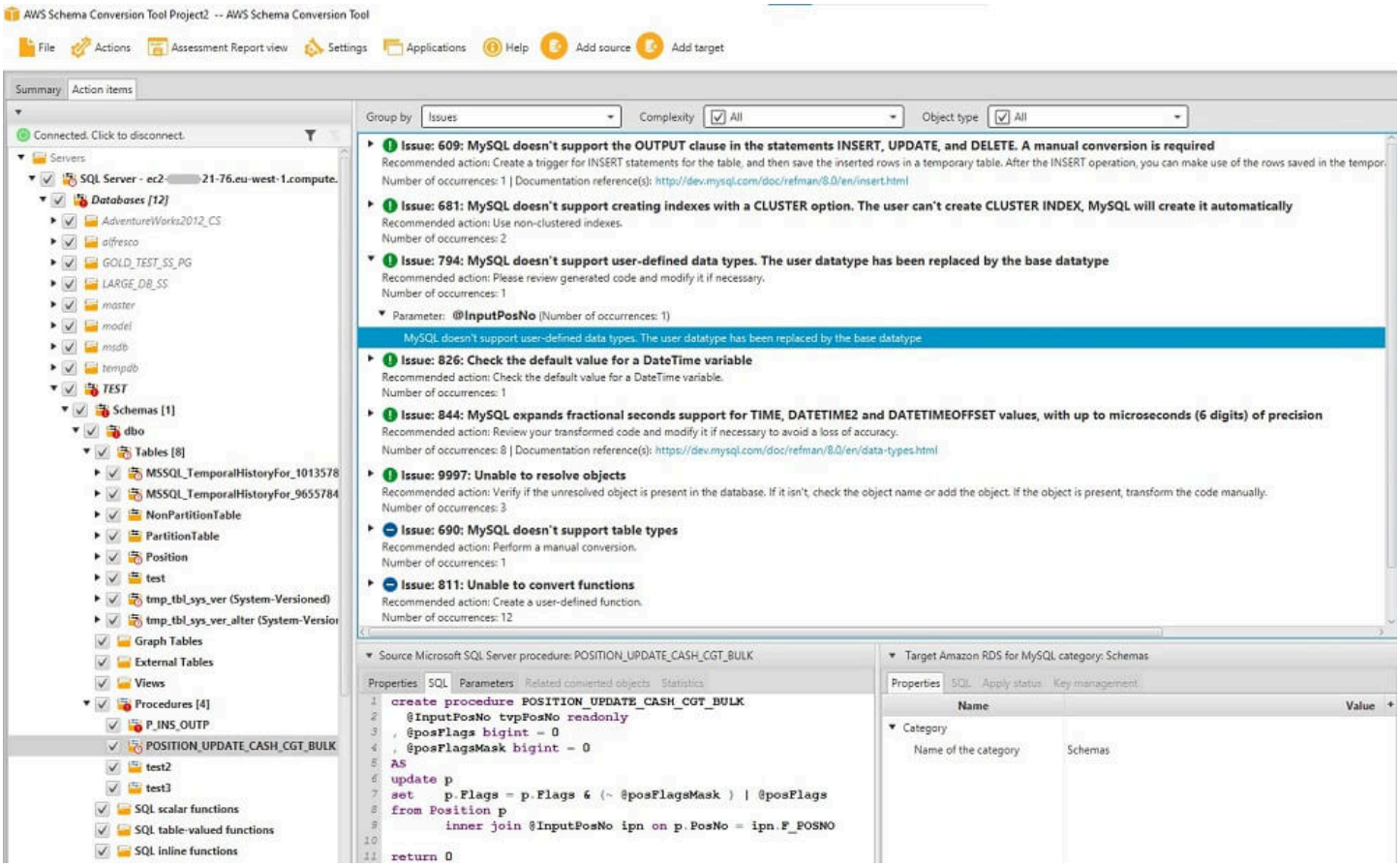
If choose Amazon RDS for Oracle as your migration target, please follow the abovementioned steps to continue to use these features on the target database after migration completes.

Elementos de acción del informe de evaluación

La vista del informe de evaluación incluye también una pestaña Elementos de acción. Esta pestaña contiene una lista de los elementos que no se pueden convertir automáticamente en el motor de base de datos de su instancia de base de datos de Amazon RDS de destino. Si selecciona un elemento de acción de la lista, AWS SCT resalta el elemento en el esquema al que se aplica el elemento de acción.

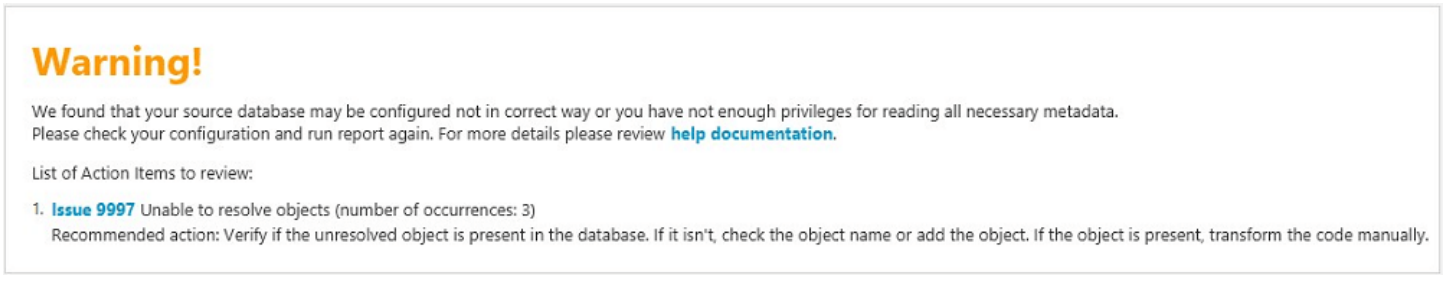
El informe también contiene recomendaciones sobre cómo convertir el elemento del esquema de forma manual. Por ejemplo, una vez finalizada la evaluación, los informes detallados de la

base de datos o el esquema muestran el esfuerzo necesario para diseñar e implementar las recomendaciones para convertir los elementos de acción. Para obtener más información sobre decidir cómo administrar las conversiones manuales, consulte Administrar conversiones manuales en AWS SCT.



Mensaje de advertencia del informe de evaluación

Para evaluar la complejidad de la conversión a otro motor de base de datos, AWS SCT necesita acceder a los objetos de la base de datos de origen. Cuando SCT no puede realizar una evaluación porque se han detectado problemas durante la digitalización, aparece un mensaje de advertencia que indica que el porcentaje de conversión general se ha reducido.



Los siguientes son los motivos por los que AWS SCT podría encontrar problemas durante el análisis:

- La cuenta de usuario conectada a la base de datos no tiene acceso a todos los objetos necesarios.
- Un objeto citado en el esquema ya no existe en la base de datos.
- SCT está intentando evaluar un objeto que está cifrado.

Para obtener más información sobre los privilegios y permisos de seguridad requeridos por SCT para su base de datos, consulte Orígenes para AWS SCT, donde encontrará la sección correspondiente a la base de datos de origen de esta guía.

Guardar el informe de evaluación

Tras crear un informe de evaluación de la migración de la base de datos, puede guardar una copia local del informe en forma de archivo PDF o archivo de valores separados por comas (CSV).

Para guardar un informe de evaluación de la migración de la base de datos como archivo PDF

1. En el menú superior, seleccione Ver y, a continuación, seleccione Ver informe de evaluación.
2. Haga clic en la pestaña Resumen.
3. Seleccione Guardar en PDF en la esquina superior derecha.

Para guardar un informe de evaluación de la migración de la base de datos como archivo CSV

1. En el menú superior, seleccione Ver y, a continuación, seleccione Ver informe de evaluación.
2. Haga clic en la pestaña Resumen.
3. Seleccione Guardar en CSV en la esquina superior derecha.

El archivo PDF contiene el resumen de acciones e información de los elementos de acción, tal y como se muestra en el siguiente ejemplo.

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 identifed 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.

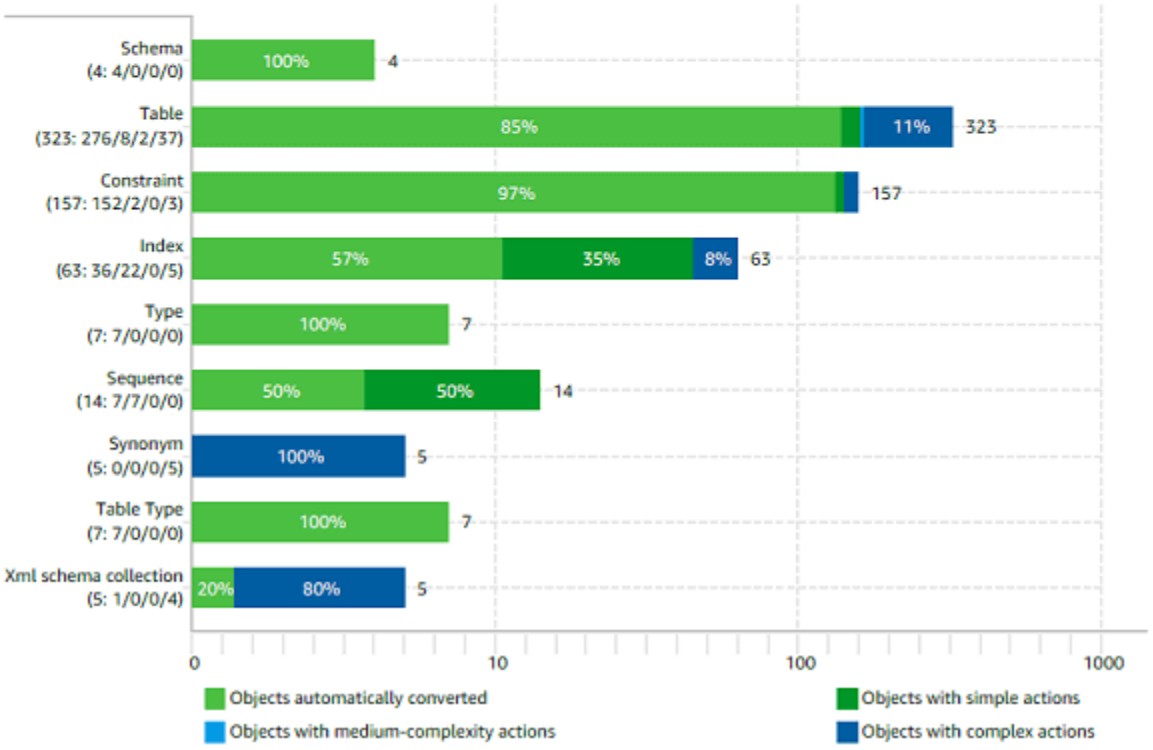
We found 7 encrypted object(s).

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



Al elegir la opción Guardar en CSV, AWS SCT crea tres archivos CSV.

El primer archivo CSV contiene la siguiente información acerca de los elementos de acción:

- Categoría
- Aparición: el nombre del archivo, el número de línea y la posición del elemento
- Número de elementos de acción
- Asunto
- Grupo

- Descripción
- Referencias de documentación
- Acción recomendada
- Complejidad estimada

El segundo archivo CSV incluye el sufijo `Action_Items_Summary` en su nombre y contiene la información sobre el número de veces que se producen todos los elementos de acción.

En el siguiente ejemplo, los valores de la columna Esfuerzo de la curva de aprendizaje indican la cantidad de esfuerzo necesaria para diseñar un método para convertir cada elemento de acción. Los valores de la columna Esfuerzo para convertir una aparición del elemento de acción indican el esfuerzo necesario para convertir cada elemento de acción, siguiendo el método diseñado. Los valores que indican el nivel de esfuerzo requerido se basan en una escala ponderada, que va desde el nivel más bajo (mínimo) hasta el más alto (máximo).

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

El tercer archivo CSV incluye `Summary` en su nombre y contiene la siguiente información:

- Categoría
- Número de objetos
- Objetos convertidos automáticamente
- Objetos con acciones sencillas
- Objetos con acciones de complejidad media
- Objetos con acciones complejas
- Total de líneas de código

Configurar el informe de evaluación

Puede configurar la cantidad de detalles que AWS SCT incluye en los informes de evaluación.

Para configurar un informe de evaluación de la migración de la base de datos

1. En el menú Configuración, seleccione Configuración global y, a continuación, seleccione Informe de evaluación.
2. En Apariciones de elementos de acción, seleccione Solo los cinco primeros problemas para limitar el número de elementos de acción de un solo tipo en el informe de evaluación. Seleccione Todos los problemas para incluir todos los elementos de acción de cada tipo en el informe de evaluación.
3. En Archivos analizados con scripts SQL, elija Indicar no más de ***X*** archivos para limitar el número de archivos de scripts SQL en el informe de evaluación*X*. Introduzca el número de archivos. Seleccione Indicar todos los archivos analizados para incluir todos los archivos de scripts SQL en el informe de evaluación.
4. Seleccione Abrir informes después de guardarlos para abrir automáticamente el archivo después de guardar una copia local del informe de evaluación de la migración de la base de datos. Para obtener más información, consulte

Tras crear un informe de evaluación de la migración de la base de datos, puede guardar una copia local del informe en forma de archivo PDF o archivo de valores separados por comas (CSV).

Para guardar un informe de evaluación de la migración de la base de datos como archivo PDF

1. En el menú superior, seleccione Ver y, a continuación, seleccione Ver informe de evaluación.
2. Haga clic en la pestaña Resumen.
3. Seleccione Guardar en PDF en la esquina superior derecha.

Para guardar un informe de evaluación de la migración de la base de datos como archivo CSV

1. En el menú superior, seleccione Ver y, a continuación, seleccione Ver informe de evaluación.
2. Haga clic en la pestaña Resumen.
3. Seleccione Guardar en CSV en la esquina superior derecha.

El archivo PDF contiene el resumen de acciones e información de los elementos de acción, tal y como se muestra en el siguiente ejemplo.

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 identifed 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.

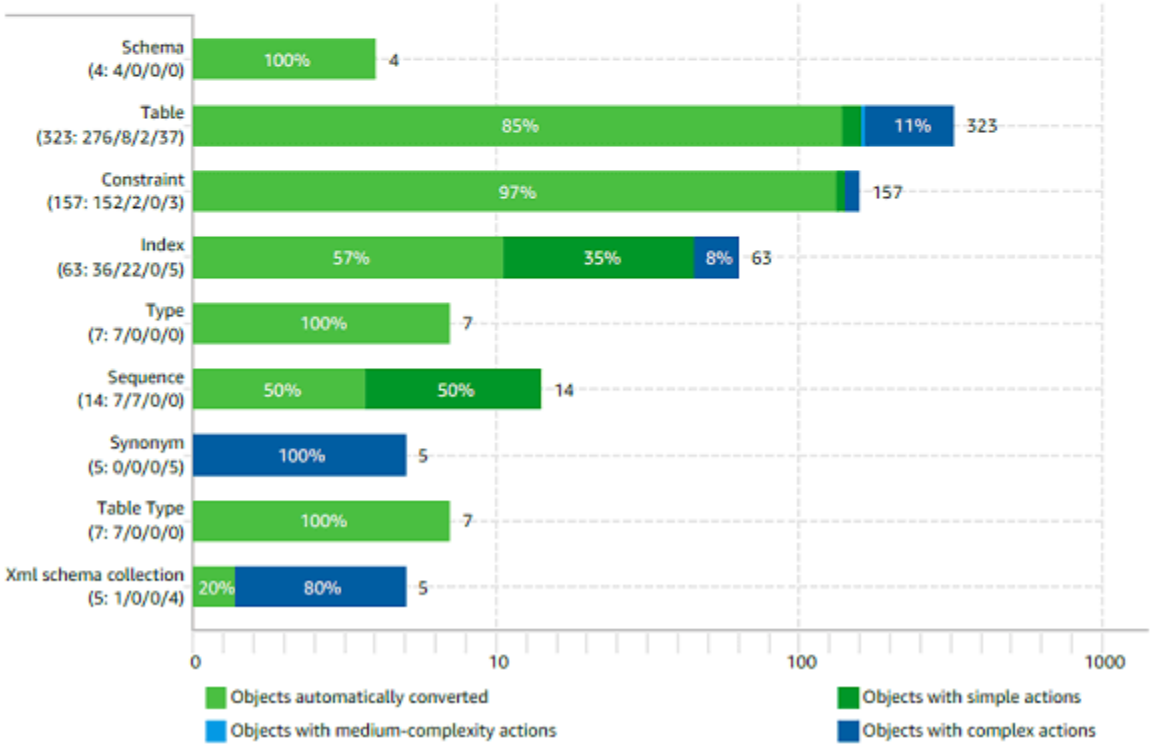
We found 7 encrypted object(s).

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



Al elegir la opción Guardar en CSV, AWS SCT crea tres archivos CSV.

El primer archivo CSV contiene la siguiente información acerca de los elementos de acción:

- Categoría
- Aparición: el nombre del archivo, el número de línea y la posición del elemento

- Número de elementos de acción

- Asunto
- Grupo
- Descripción
- Referencias de documentación
- Acción recomendada
- Complejidad estimada

El segundo archivo CSV incluye el sufijo `Action_Items_Summary` en su nombre y contiene la información sobre el número de veces que se producen todos los elementos de acción.

En el siguiente ejemplo, los valores de la columna Esfuerzo de la curva de aprendizaje indican la cantidad de esfuerzo necesaria para diseñar un método para convertir cada elemento de acción. Los valores de la columna Esfuerzo para convertir una aparición del elemento de acción indican el esfuerzo necesario para convertir cada elemento de acción, siguiendo el método diseñado. Los valores que indican el nivel de esfuerzo requerido se basan en una escala ponderada, que va desde el nivel más bajo (mínimo) hasta el más alto (máximo).

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

El tercer archivo CSV incluye `Summary` en su nombre y contiene la siguiente información:

- Categoría
- Número de objetos
- Objetos convertidos automáticamente
- Objetos con acciones sencillas
- Objetos con acciones de complejidad media
- Objetos con acciones complejas

- Total de líneas de código

.

Crear un informe de evaluación multiservidor para la migración de la base de datos

Determine la mejor dirección de destino para su entorno general, cree un informe de evaluación multiservidor.

Un informe de evaluación multiservidor evalúa varios servidores en función de los datos que proporcione para cada definición de esquema que desee evaluar. La definición del esquema contiene los parámetros de conexión al servidor de la base de datos y el nombre completo de cada esquema. Tras evaluar cada esquema, AWS SCT elabora un informe de evaluación resumido y agregado para la migración de la base de datos entre varios servidores. Este informe muestra la complejidad estimada de cada posible objetivo de migración.

Puede utilizar AWS SCT para crear un informe de evaluación multiservidor para las siguientes bases de datos de origen y destino.

Base de datos de origen	Bases de datos de destino
Amazon Redshift	Amazon Redshift
Base de datos SQL Azure	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
Azure Synapse Analytics	Amazon Redshift
BigQuery	Amazon Redshift
Greenplum	Amazon Redshift
IBM Db2 para z/OS	Amazon Aurora MySQL-Compatible Edition (Aurora MySQL), Amazon Aurora PostgreSQL-Compatible Edition (Aurora PostgreSQL), MySQL, PostgreSQL

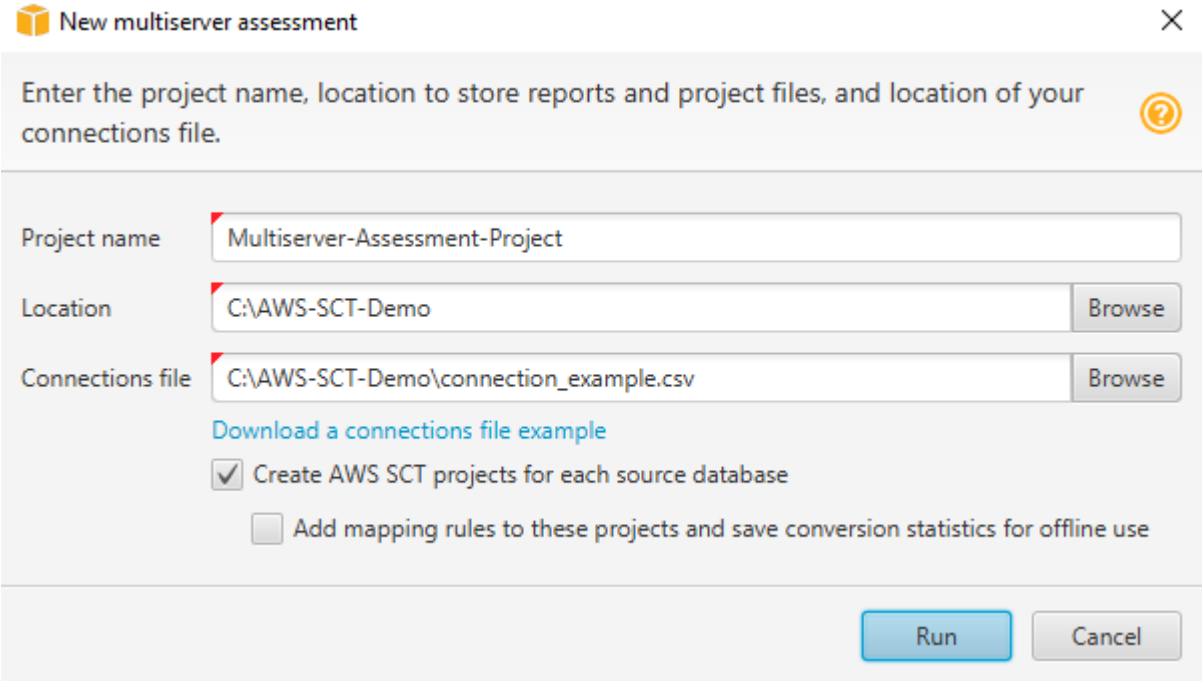
Base de datos de origen	Bases de datos de destino
IBM Db2 LUW	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Microsoft SQL Server	Aurora MySQL, Aurora PostgreSQL, Amazon Redshift, Babelfish para Aurora PostgreSQL, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL	Aurora PostgreSQL, MySQL, PostgreSQL
Netezza	Amazon Redshift
Oracle	Aurora MySQL, Aurora PostgreSQL, Amazon Redshift, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
SAP ASE	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Snowflake	Amazon Redshift
Teradata	Amazon Redshift
Vertica	Amazon Redshift

Realizar una evaluación multiservidor

Utilice el siguiente procedimiento para realizar una evaluación multiservidor con AWS SCT. No necesita crear un proyecto nuevo en AWS SCT para realizar una evaluación multiservidor. Antes de empezar, prepare un archivo de valores separados por comas (CSV) con los parámetros de conexión a la base de datos. Además, instale todos los controladores de base de datos necesarios y establezca la ubicación de los controladores en la configuración de AWS SCT. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios.

Para realizar una evaluación multiservidor y crear un informe resumido agregado

1. En AWS SCT, elija Archivo, Evaluación multiservidor nueva. Se abre el cuadro de diálogo Evaluación multiservidor nueva.



2. Seleccione Descargar un ejemplo de archivo de conexiones para descargar una plantilla vacía de un archivo CSV con los parámetros de conexión a la base de datos.
3. Introduzca los valores para Nombre del proyecto, Ubicación (para almacenar los informes) y Archivo de conexiones (un archivo CSV).
4. Elija Crear proyectos de AWS SCT para cada base de datos de origen para crear automáticamente los proyectos de migración después de generar el informe de evaluación.
5. Con la opción Crear proyectos de AWS SCT para cada base de datos de origen activada, puede elegir Agregar reglas de asignación a estos proyectos y guardar las estadísticas de conversión para utilizarlas sin conexión. En este caso, AWS SCT agregará reglas de asignación a cada proyecto y guardará los metadatos de la base de datos de origen en el proyecto. Para obtener más información, consulte Ejecutar AWS SCT en el modo sin conexión.
6. Elija Ejecutar.

   Aparece una barra de progreso que indica el ritmo de la evaluación de la base de datos. La cantidad de motores de destino puede afectar al tiempo de ejecución de la evaluación.
7. Elija Sí si aparece el siguiente mensaje: El análisis completo de todos los servidores de bases de datos puede tardar algún tiempo. ¿Desea continuar?

Una vez elaborado el informe de evaluación multiservidor, aparecerá una pantalla que lo indica.

8. Elija Abrir informe para ver el informe de evaluación resumido agregado.

De forma predeterminada, AWS SCT genera un informe agregado para todas las bases de datos de origen y un informe de evaluación detallado para cada nombre de esquema de una base de datos de origen. Para obtener más información, consulte Localizar y ver informes.

Con la opción Crear proyectos de AWS SCT para cada base de datos de origen activada, AWS SCT crea un proyecto vacío para cada base de datos de origen. AWS SCT también crea informes de evaluación tal como se describió anteriormente. Tras analizar estos informes de evaluación y elegir el destino de migración para cada base de datos de origen, agregue las bases de datos de destino a estos proyectos vacíos.

Con la opción Agregar reglas de asignación a estos proyectos y guardar estadísticas de conversión para usar sin conexión activada, AWS SCT crea un proyecto para cada base de datos de origen. En estos proyectos se incluye la siguiente información:

- Su base de datos de origen y una plataforma de base de datos de destino virtual. Para obtener más información, consulte Usar destinos virtuales.
- Una regla de asignación para este par origen-destino. Para obtener más información, consulte Creación de reglas de asignación.
- Un informe de evaluación de la migración de la base de datos para este par origen-destino.
- Metadatos del esquema de origen, que le permiten utilizar este proyecto de AWS SCT sin conexión. Para obtener más información, consulte Ejecutar AWS SCT en el modo sin conexión.

Preparar un archivo CSV de entrada

Para proporcionar los parámetros de conexión como entrada para el informe de evaluación multiservidor, utilice un archivo CSV como se muestra en el ejemplo siguiente.

```
Name,Description,Secret Manager Key,Server IP,Port,Service Name,Database name,BigQuery
 path,Source Engine,Schema Names,Use Windows Authentication,Login,Password,Use
 SSL,Trust store,Key store,SSL authentication,Target Engines
Sales,,,192.0.2.0,1521,pdb,,,ORACLE,Q4_2021;FY_2021,,user,password,,,,,POSTGRESQL;AURORA_POSTGR
Marketing,,,ec2-a-b-c-d.eu-
west-1.compute.amazonaws.com,1433,,target_audience,,MSSQL,customers.dbo,,user,password,,,,,AURO
HR,,,192.0.2.0,1433,,employees,,MSSQL,employees.%,true,,,,,,,AURORA_POSTGRESQL
```

```
Customers,,secret-name,,,,,MYSQL,customers,,,,,,,,AURORA_POSTGRESQL
Analytics,,,198.51.100.0,8195,,STATISTICS,,DB2LUW,BI_REPORTS,,user,password,,,,,POSTGRESQL
Products,,,203.0.113.0,8194,,,,TERADATA,new_products,,user,password,,,,,REDSHIFT
```

En el ejemplo anterior, se utiliza un punto y coma para separar los dos nombres de esquema de la base de datos `Sales`. También se utiliza un punto y coma para separar las dos plataformas de migración de bases de datos de destino de la base de datos `Sales`.

Además, en el ejemplo anterior se utiliza AWS Secrets Manager para conectar a la base de datos `Customers` y la autenticación de Windows para conectar a la base de datos `HR`.

Puede crear un archivo CSV nuevo o descargar una plantilla para un archivo CSV desde AWS SCT y completar la información requerida. Asegúrese de que la primera fila del archivo CSV incluya los mismos nombres de columna que se muestran en el ejemplo anterior.

Para descargar una plantilla del archivo CSV de entrada

1. Inicie AWS SCT.
2. En Archivo, seleccione Evaluación multiservidor nueva.
3. Elija Descargar un ejemplo de archivo de conexiones.

Asegúrese de que el archivo CSV incluya los siguientes valores, proporcionados por la plantilla:

- Nombre: la etiqueta de texto que ayuda a identificar la base de datos. AWS SCT muestra esta etiqueta de texto en el informe de evaluación.
- Descripción: un valor opcional en el que puede proporcionar información adicional sobre la base de datos.
- Clave de Secrets Manager: el nombre del secreto que almacena las credenciales de la base de datos en AWS Secrets Manager. Para usar Secrets Manager, guarde los perfiles de AWS en AWS SCT. Para obtener más información, consulte Utilizar AWS Secrets Manager.

  > **Important**
  >
  > AWS SCT ignora el parámetro Clave de Secrets Manager si incluye los parámetros IP del servidor, Puerto, Nombre de usuario y Contraseña en el archivo de entrada.

- IP de servidor: escriba el nombre del servicio de nombres de dominio (DNS) o la dirección IP del servidor de la base de datos de origen.

- Puerto: el puerto utilizado para conectarse al servidor de su base de datos de origen.
- Nombre del servicio: si utiliza un nombre de servicio para conectarse a la base de datos de Oracle, el nombre del servicio de Oracle al que se va a conectar.
- Nombre de la base de datos: el nombre de la base de datos. Para las bases de datos de Oracle, utilice el ID de sistema de Oracle (SID).
- Ruta de BigQuery: la ruta al archivo de claves de la cuenta de servicio de su base de datos de BigQuery de origen. Para obtener más información sobre la creación de este archivo, consulte Privilegios para BigQuery como origen.
- Motor de origen: el tipo de base de datos de origen. Utilice uno de los siguientes valores:
  - AZURE_MSSQL para una base de datos de Azure SQL.
  - AZURE_SYNAPSE para una base de datos de Azure Synapse Analytics.
  - GOOGLE_BIGQUERY para una base de datos de BigQuery.
  - DB2ZOS para una base de datos de IBM Db2 para z/OS.
  - DB2LUW para una base de datos de IBM Db2 LUW.
  - GREENPLUM para una base de datos de Greenplum.
  - MSSQL para una base de datos de Microsoft SQL Server.
  - MYSQL para una base de datos de MySQL.
  - NETEZZA para una base de datos de Netezza.
  - ORACLE para una base de datos de Oracle.
  - POSTGRESQL para una base de datos de PostgreSQL.
  - REDSHIFT para una base de datos de Amazon Redshift.
  - SNOWFLAKE para una base de datos de Snowflake.
  - SYBASE_ASE para una base de datos de SAP ASE.
  - TERADATA para una base de datos de Teradata.
  - VERTICA para una base de datos de Vertica.
- Nombres de esquemas: los nombres de los esquemas de bases de datos que se van a incluir en el informe de evaluación.

  Para Azure SQL Database, Azure Synapse Analytics, BigQuery, Netezza, SAP ASE, Snowflake y SQL Server, use el siguiente formato de nombre de esquema:

  db_name.schema_name

Sustituya *db_name* por el nombre de la base de datos de origen.

Sustituya *schema_name* por el nombre del esquema de origen.

Incluya los nombres de las bases de datos o los esquemas que incluyan un punto entre comillas dobles como se muestra a continuación: `"database.name"."schema.name"`.

Separe los nombres de varios esquemas mediante punto y coma como se muestra a continuación: `Schema1;Schema2`.

Los nombres de las base de datos y los esquemas distinguen entre mayúsculas y minúsculas.

Utilice el porcentaje (`%`) como carácter comodín para reemplazar los símbolos del nombre de la base de datos o del esquema. El ejemplo anterior utiliza el porcentaje (`%`) como carácter comodín para incluir todos los esquemas de la base de datos `employees` en el informe de evaluación.

- Usar autenticación de Windows: si utiliza la autenticación de Windows para conectarse a la base de datos de Microsoft SQL Server, introduzca true. Para obtener más información, consulte Uso de la autenticación de Windows al utilizar Microsoft SQL Server como origen.
- Nombre de usuario: el nombre de usuario para conectarse a su servidor de base de datos de origen.
- Contraseña: la contraseña para conectarse al servidor de base de datos de origen.
- Usar SSL: si utiliza una capa de sockets seguros (SSL) para conectarse a la base de datos de origen, introduzca true.
- Almacén de confianza: el almacén de confianza que se utilizará para la conexión SSL.
- Almacén de confianza: el almacén de claves que se utilizará para la conexión SSL.
- Autenticación SSL: si utiliza la autenticación SSL mediante certificado, introduzca true.
- Motores de destino: las plataformas de bases de datos de destino. Utilice los siguientes valores para especificar uno o más destinos en el informe de evaluación:
  - AURORA_MYSQL para una base de datos compatible con Aurora MySQL.
  - AURORA_POSTGRESQL para una base de datos compatible con Aurora PostgreSQL.
  - BABELFISH: para una base de datos de Babelfish para Aurora PostgreSQL.
  - MARIA_DB para una base de datos de MariaDB.
  - MSSQL para una base de datos de Microsoft SQL Server.
  - MYSQL para una base de datos de MySQL.

- ORACLE para una base de datos de Oracle.
- POSTGRESQL para una base de datos de PostgreSQL.
- REDSHIFT para una base de datos de Amazon Redshift.

Separe varios destinos con punto y coma como en este ejemplo: `MYSQL;MARIA_DB`. La cantidad de destinos afecta al tiempo que se tarda en ejecutar la evaluación.

Localizar y ver informes

La evaluación multiservidor genera dos tipos de informes:

- Un informe agregado de todas las bases de datos de origen.
- Un informe de evaluación detallado de las bases de datos de destino para cada nombre de esquema de una base de datos de origen.

Los informes se almacenan en el directorio que haya elegido como Ubicación en el cuadro de diálogo Evaluación multiservidor nueva.

Para acceder a los informes detallados, puede navegar por los subdirectorios, que están organizados por base de datos de origen, nombre de esquema y motor de base de datos de destino.

Los informes agregados muestran información en cuatro columnas sobre la complejidad de conversión de una base de datos de destino. Las columnas incluyen información sobre la conversión de objetos de código, objetos de almacenamiento, elementos de sintaxis y complejidad de conversión.

El siguiente ejemplo muestra información para la conversión de dos esquemas de bases de datos de Oracle a Amazon RDS para PostgreSQL.

Server IP address and port	Secret Manager key	Name	Description	Database name	Schema name	Code object conversion % for "Amazon RDS for PostgreSQL"	Storage object conversion % for "Amazon RDS for PostgreSQL"	Syntax Elements conversion % for "Amazon RDS for PostgreSQL"	Conversion Complexity for "Amazon RDS for PostgreSQL"
192.0.2.0:1521		Sales		ORCL	Q4_2021	97.78%	100.00%	98.76%	1
192.0.2.0:1521		Sales		pdb	FY_2021	82.35%	85.19%	99.24%	10

Las mismas cuatro columnas se adjuntan a los informes para cada motor de base de datos de destino adicional especificado.

Para obtener información detallada acerca de la forma de leer esta información, consulte a continuación.

Resultado de un informe de evaluación agregado

El informe agregado de evaluación de la migración de bases de datos multiservidor en AWS Schema Conversion Tool es un archivo CSV con las siguientes columnas:

- `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`*

Para recopilar información, AWS SCT ejecuta informes de evaluación completos y, a continuación, agrega los informes por esquemas.

En el informe, los tres campos siguientes muestran el porcentaje de conversión automática posible en función de la evaluación:

% de conversión de objetos de código

El porcentaje de objetos de código del esquema que AWS SCT puede convertir automáticamente o con cambios mínimos. Los objetos de código incluyen procedimientos, funciones, vistas y similares.

% de conversión de objetos de almacenamiento

El porcentaje de objetos de almacenamiento que SCT puede convertir automáticamente o con cambios mínimos. Los objetos de almacenamiento incluyen tablas, índices, restricciones y similares.

% de conversión de elementos de sintaxis

El porcentaje de elementos de sintaxis que SCT puede convertir automáticamente. Los elementos de sintaxis incluyen cláusulas `SELECT`, `FROM`, `DELETE` y `JOIN` y similares.

El cálculo de la complejidad de la conversión se basa en la noción de elementos de acción. Un elemento de acción refleja un tipo de problema encontrado en el código fuente que hay que solucionar manualmente durante la migración a un destino concreto. Un elemento de acción puede tener varias apariciones.

Una escala ponderada identifica el nivel de complejidad necesario para realizar una migración. El número 1 representa el nivel más bajo de complejidad y el número 10 representa el nivel más alto de complejidad.

Conversión de esquemas de base de datos mediante AWS SCT

Puede utilizar la AWS Schema Conversion Tool (AWS SCT) para convertir sus esquemas de base de datos existente de un motor de base de datos a otro. La conversión de una base de datos con la interfaz de usuario de AWS SCT puede ser bastante sencillo, pero hay varios aspectos que se deben tener en cuenta antes de realizar la conversión.

Por ejemplo, puede utilizar AWS SCT para hacer lo siguiente:

- También puede utilizar AWS SCT para copiar un esquema de base de datos en las instalaciones existente a una instancia de base de datos de Amazon RDS que ejecute el mismo motor. Puede utilizar esta característica para analizar posibles ahorros de costos en la migración a la nube y en el cambio del tipo de licencia.
- En algunos casos, las características de la base de datos de origen no se pueden convertir a características de Amazon RDS equivalentes. Si aloja y autoadministra una base de datos en la plataforma de Amazon Elastic Compute Cloud (Amazon EC2), puede simular estas características sustituyendo los servicios de AWS por ellas.
- AWS SCT automatiza gran parte del proceso de conversión de su esquema de base de datos de procesamiento de transacciones online (OLTP) a una instancia de base de datos MySQL de Amazon Relational Database Service (Amazon RDS), un clúster de base de datos de Amazon Aurora o una instancia de base de datos PostgreSQL. Los motores de base de datos de origen y de destino contienen diferentes características y capacidades, por lo que AWS SCT intenta crear un esquema equivalente en su instancia de base de datos de Amazon RD siempre que sea posible. Si no es posible la conversión directa, la AWS SCT proporciona una lista de posibles acciones que puede realizar.

Temas

- Crear reglas de migración en AWS SCT
- Convertir su esquema con AWS SCT
- Administrar conversiones manuales en AWS SCT
- Actualizar un esquema convertido en AWS SCT
- Guardar y aplicar un esquema convertido en AWS SCT
- Comparación de esquemas de bases de datos

- Buscar objetos transformados relacionados

AWS SCT admite las siguientes conversiones de procesamiento de transacciones en línea (OLTP).

Base de datos de origen	Bases de datos de destino
IBM Db2 para z/OS (versión 12)	Amazon Aurora MySQL-Compatible Edition (Aurora MySQL), Amazon Aurora PostgreSQL-Compatible Edition (Aurora PostgreSQL), MySQL, PostgreSQL
IBM Db2 LUW (versiones 9.1, 9.5, 9.7, 10.5, 11.1 y 11.5)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL
Microsoft Azure SQL Database	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
Microsoft SQL Server (versión 2008 R2 y posteriores)	Aurora MySQL, Aurora PostgreSQL, Babelfish para Aurora PostgreSQL, MariaDB, Microsoft SQL Server, MySQL, PostgreSQL
MySQL (versión 5.5 y posteriores)	Aurora PostgreSQL, MySQL, PostgreSQL Puede migrar esquemas y datos de MySQL a un clúster de base de datos de Aurora MySQL sin usar AWS SCT. Para obtener más información, consulte Migrar datos a un clúster de base de datos de Amazon Aurora.
Oracle (versión 10.2 y posteriores)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, Oracle, PostgreSQL
PostgreSQL (versión 9.1 y posterior)	Aurora MySQL, Aurora PostgreSQL, MySQL, PostgreSQL
SAP ASE (12.5, 15.0, 15.5, 15.7 y 16.0)	Aurora MySQL, Aurora PostgreSQL, MariaDB, MySQL, PostgreSQL

Para obtener más información acerca de la conversión de un esquema de almacenamiento de datos, consulte Convertir esquemas de almacenamiento de datos a Amazon Redshift mediante AWS SCT.

Para convertir su esquema de base de datos a Amazon RDS, siga estos pasos de alto nivel:

- Crear reglas de migración en AWS SCT: antes de convertir su esquema con AWS SCT, puede configurar reglas que cambien el tipo de datos de las columnas, mover objetos de un esquema a otro y cambiar los nombres de los objetos.
- Convertir su esquema con AWS SCT: AWS SCT crea una versión local del esquema convertido para que lo revise, pero no se aplica a su instancia de base de datos de destino hasta que esté listo.
- Crear informes de evaluación de migración con AWS SCT: AWS SCT crea un informe de evaluación de la migración de la base de datos en el que se detallen los elementos del esquema que no se pueden convertir automáticamente. Puede utilizar este informe para identificar dónde necesita crear un esquema en su instancia de base de datos de Amazon RDS que sea compatible con su base de datos de origen.
- Administrar conversiones manuales en AWS SCT: si tiene elementos del esquema que no se pueden convertir automáticamente, dispone de dos opciones: actualizar el esquema de origen y, a continuación, volver a convertirlo, o bien crear elementos de esquema equivalentes en su instancia de base de datos de Amazon RDS de destino.
- Actualizar un esquema convertido en AWS SCT: puede actualizar el proyecto de AWS SCT con el esquema más reciente de su base de datos de origen.
- Guardar y aplicar un esquema convertido en AWS SCT: cuando esté listo, haga que AWS SCT aplique el esquema convertido en su proyecto local a su instancia de base de datos de Amazon RDS de destino.

Crear reglas de migración en AWS SCT

Antes de convertir el esquema con AWS SCT, puede configurar las reglas de migración. Las reglas de migración en AWS SCT pueden realizar transformaciones tales como cambiar el tipo de datos de las columnas, mover objetos de un esquema a otro y cambiar los nombres de los objetos. Por ejemplo, suponga que tiene un conjunto de tablas en el esquema de origen denominado `test_TABLE_NAME`. Puede configurar una regla que cambie el prefijo `test_` por el prefijo `demo_` del esquema de destino.

> **Note**
>
> Solo puede crear reglas de migración para motores de bases de datos de origen y destino diferentes.

Puede crear reglas de migración que realicen las siguientes tareas:

- Agregar, eliminar o reemplazar un prefijo
- Agregar, eliminar o reemplazar un sufijo
- Cambiar intercalación de columnas
- Cambiar el tipo de datos
- Cambiar la longitud de `char`, `varchar`, `nvarchar`, y los tipos de datos `string`
- Mover objetos
- Cambiar el nombre de los objetos

Puede crear reglas de migración para los siguientes objetos:

- Base de datos
- Esquema
- Tabla
- Columna

Creación de reglas de migración

Puede crear reglas de migración y guardarlas como parte de su proyecto. Con el proyecto abierto, utilice el siguiente procedimiento para crear reglas de migración.

Para crear reglas de migración

1. En el menú Ver, seleccione Vista principal.
2. En Asignaciones de servidores, elija un par de servidores de origen y de destino.
3. Elija Regla de migración nueva. Aparecerá el cuadro de diálogo Reglas de transformación.
4. Seleccione Add new rule. Se agregará una nueva fila a la lista de reglas.
5. Configure su regla:

a. En Name (Nombre), ingrese un nombre para la regla.

b. En For, seleccione el tipo de objeto al que se aplicará la regla.

c. En where, introduzca un filtro que se aplicará a los objetos antes de aplicar la regla de migración. La cláusula where se evalúa mediante una cláusula like. Puede especificar un nombre exacto para seleccionar un objeto o introducir un patrón para seleccionar varios objetos.

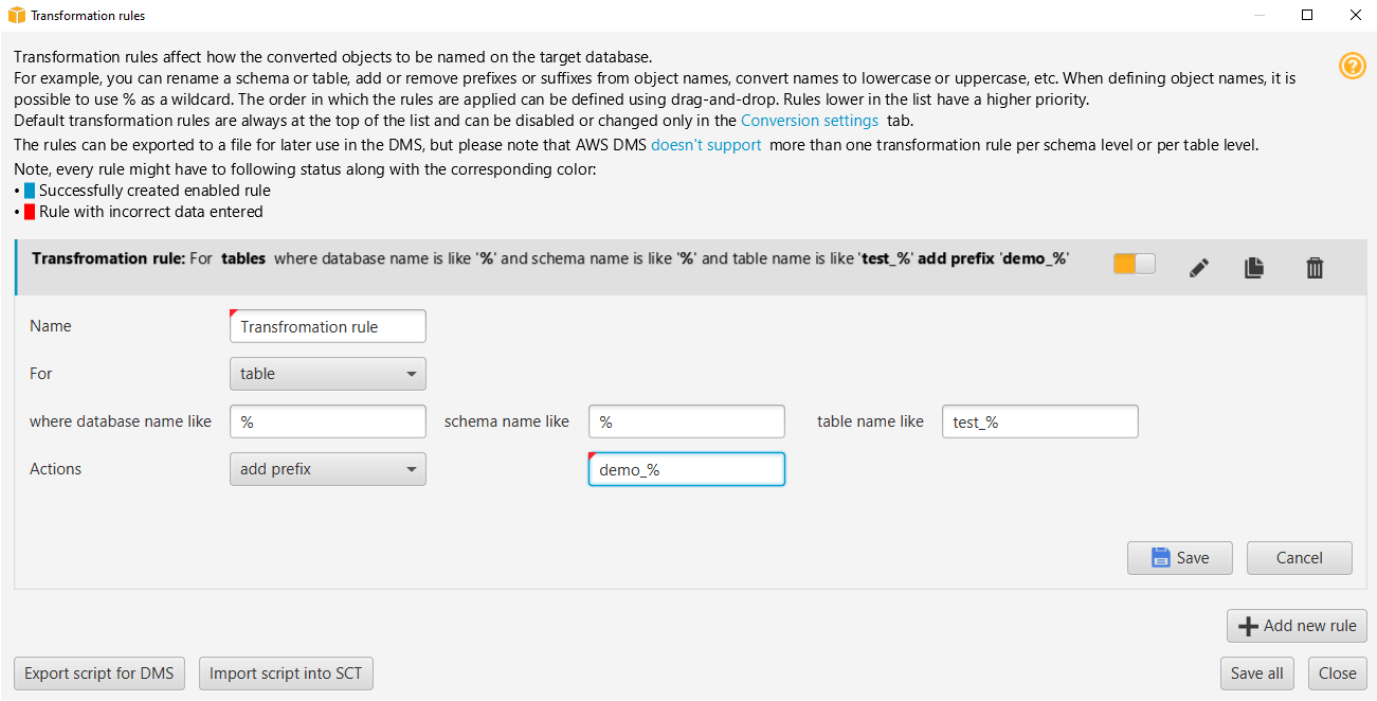
   Los campos disponibles para la cláusula where son diferentes en función del tipo de objeto. Por ejemplo, si el tipo de objeto es esquema, solo habrá un campo disponible, para el nombre del esquema.

d. En Acciones, seleccione el tipo de regla de migración que quiera crear.

e. En función del tipo de regla, introduzca uno o dos valores adicionales. Por ejemplo, para cambiar el nombre de un objeto, introduzca el nuevo nombre del objeto. Para sustituir un prefijo, introduzca el prefijo anterior y el nuevo prefijo.

   Para los tipos de datos char, varchar, nvarchar y string, puede cambiar la longitud del tipo de datos mediante el operador de multiplicación. Por ejemplo, el valor `%*4` transforma el tipo de datos `varchar(10)` en `varchar(40)`.

6. Tras haber configurado su regla de migración, seleccione Guardar para guardar su regla. También puede seleccionar Cancel para cancelar los cambios.



7. Cuando haya acabado de agregar, editar y eliminar reglas, seleccione Save All para guardar todos los cambios.
8. Seleccione Cerrar para cerrar el cuadro de diálogo Reglas de transformación.

Puede utilizar el icono de alternar para desactivar una regla de migración sin eliminarla. Puede utilizar el icono de copia para duplicar una regla de migración existente. Puede utilizar el icono del lapicero para editar una regla de migración existente. Puede utilizar el icono de eliminar para borrar una regla de migración existente. Para guardar los cambios que realice en sus reglas de migración, seleccione Guardar todo.

Exportar reglas de migración

Si utiliza AWS DMS para migrar sus datos de la base de datos de origen a la de destino, puede proporcionar información sobre sus reglas de migración a AWS DMS. Para obtener más información sobre las tareas, consulte Trabajar con tareas de replicación de AWS Database Migration Service.

Para exportar reglas de migración

1. En AWS Schema Conversion Tool, seleccione Vista de asignaciones en el menú Vista.
2. En Reglas de migración, elija una regla de migración y, a continuación, elija Modificar regla de migración.
3. Elija Exportar script para AWS DMS.
4. Seleccione la ubicación en la que desee guardar el script y después seleccione Save. Su reglas de migración se guardan como un script JSON que AWS DMS puede consumir.

Convertir su esquema con AWS SCT

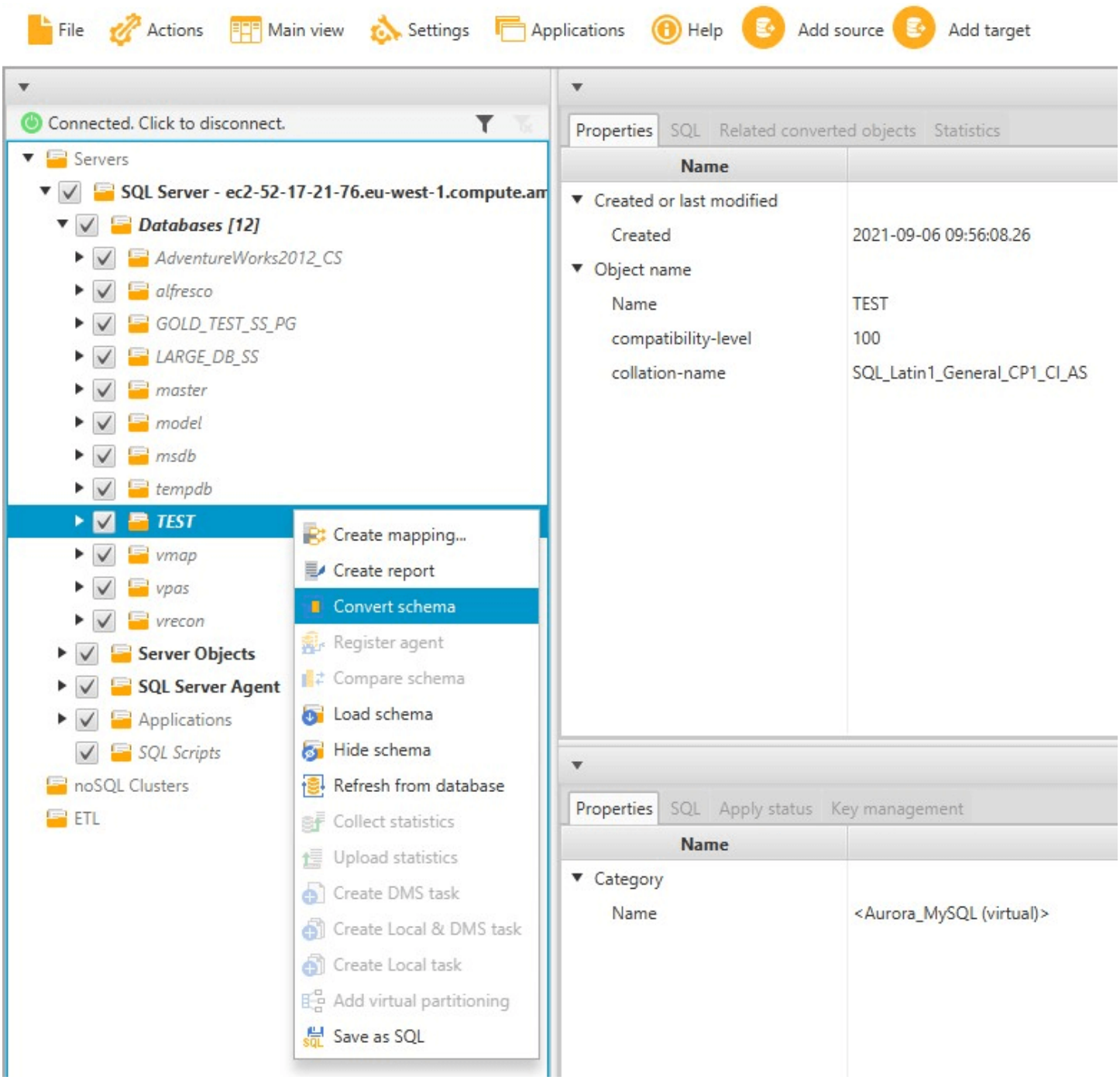
Tras haber conectado su proyecto tanto a la base de datos de origen como a la instancia de base de datos de Amazon RDS de destino, su proyecto de AWS Schema Conversion Tool mostrará el esquema de su base de datos de origen en el panel izquierdo. El esquema se presentan en un formato de vista de árbol y cada nodo del árbol se carga progresivamente. Al seleccionar un nodo en la vista de árbol, AWS SCT solicita la información del esquema de la base de datos de origen en ese momento.

Puede seleccionar elementos del esquema en su base de datos de origen y, después, convertir el esquema en un esquema equivalente para el motor de la base de datos de su instancia de base de

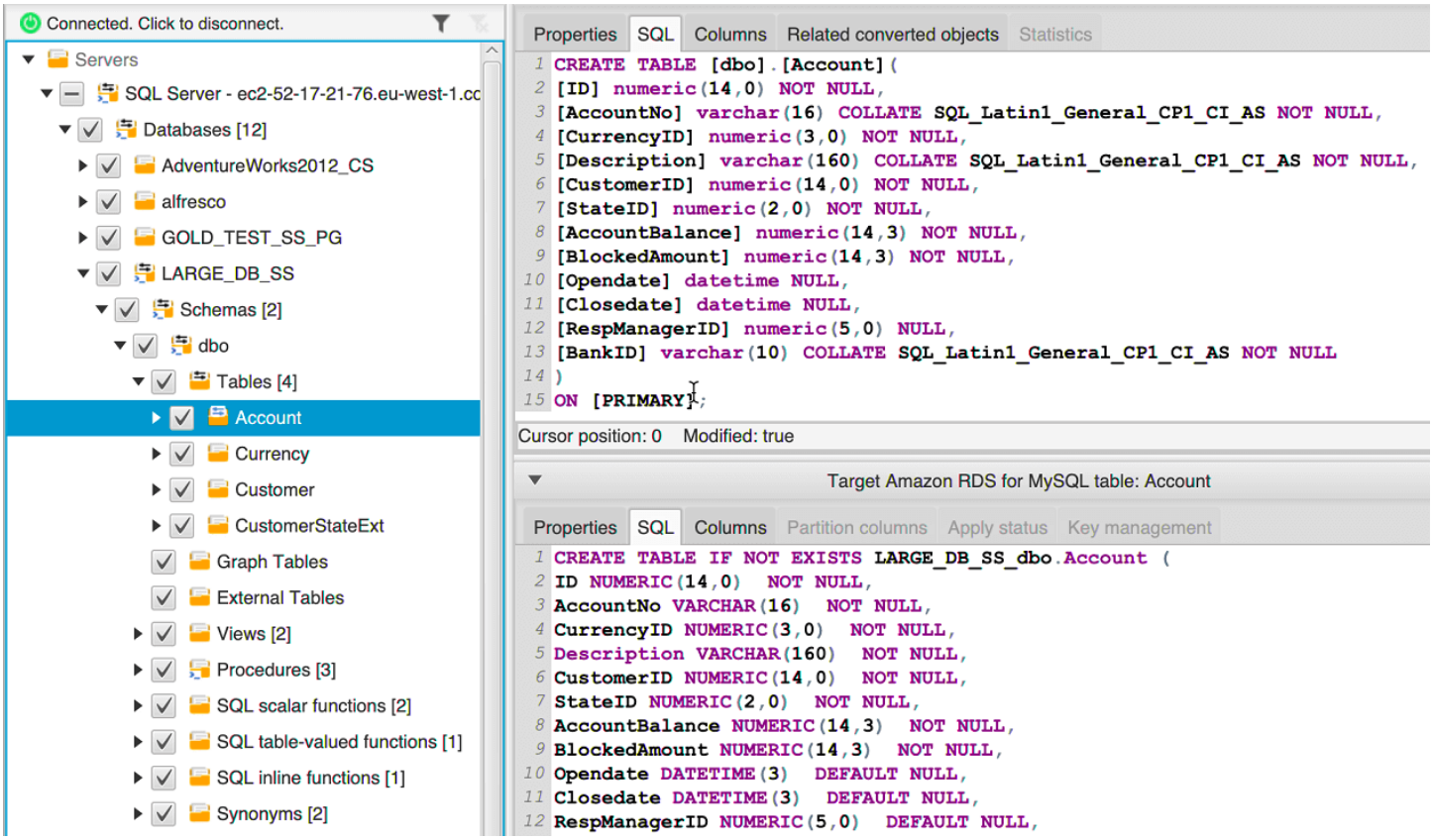
datos de destino. Puede seleccionar cualquier elemento del esquema en su base de datos de origen para convertirlo. Si el elemento del esquema que seleccione depende de un elemento principal, AWS SCT generará también el esquema para el elemento principal. Por ejemplo, suponga que elige una tabla para convertir. Si es así, AWS SCT genera el esquema de la tabla y la base de datos en la que se encuentra la tabla.

Convertir esquemas

Para convertir un esquema de la base de datos de origen, active la casilla de verificación del nombre del esquema que desee convertir. A continuación, seleccione este esquema en el panel izquierdo de su proyecto. AWS SCT resalta el nombre del esquema en azul. Abra el menú contextual (clic secundario) para el esquema y seleccione Convertir esquemas, como se muestra a continuación.



Una vez que haya convertido el esquema de su base de datos de origen, puede elegir los elementos del esquema en el panel izquierdo de su proyecto y ver el esquema convertido en los paneles centrales de su proyecto. El panel inferior central muestra las propiedades y el comando SQL para crear el esquema convertido, como se muestra a continuación.



Tras haber convertido el esquema, podrá guardar el proyecto. La información del esquema de su base de datos de origen se guarda con su proyecto. Esta funcionalidad supone que puede trabajar sin conexión sin estar conectado a su base de datos de origen. AWS SCT se conecta a su base de datos de origen para actualizar el esquema en su proyecto si selecciona Refresh from Database (Actualizar desde base de datos) para su base de datos de origen. Para obtener más información, consulte Actualizar un esquema convertido en AWS SCT.

Puede crear un informe de evaluación de la migración de la base de datos para los elementos que no se puedan convertir automáticamente. El informe de evaluación es útil para identificar y solucionar los elementos del esquema que no se puedan convertir automáticamente. Para obtener más información, consulte Crear informes de evaluación de migración con AWS SCT.

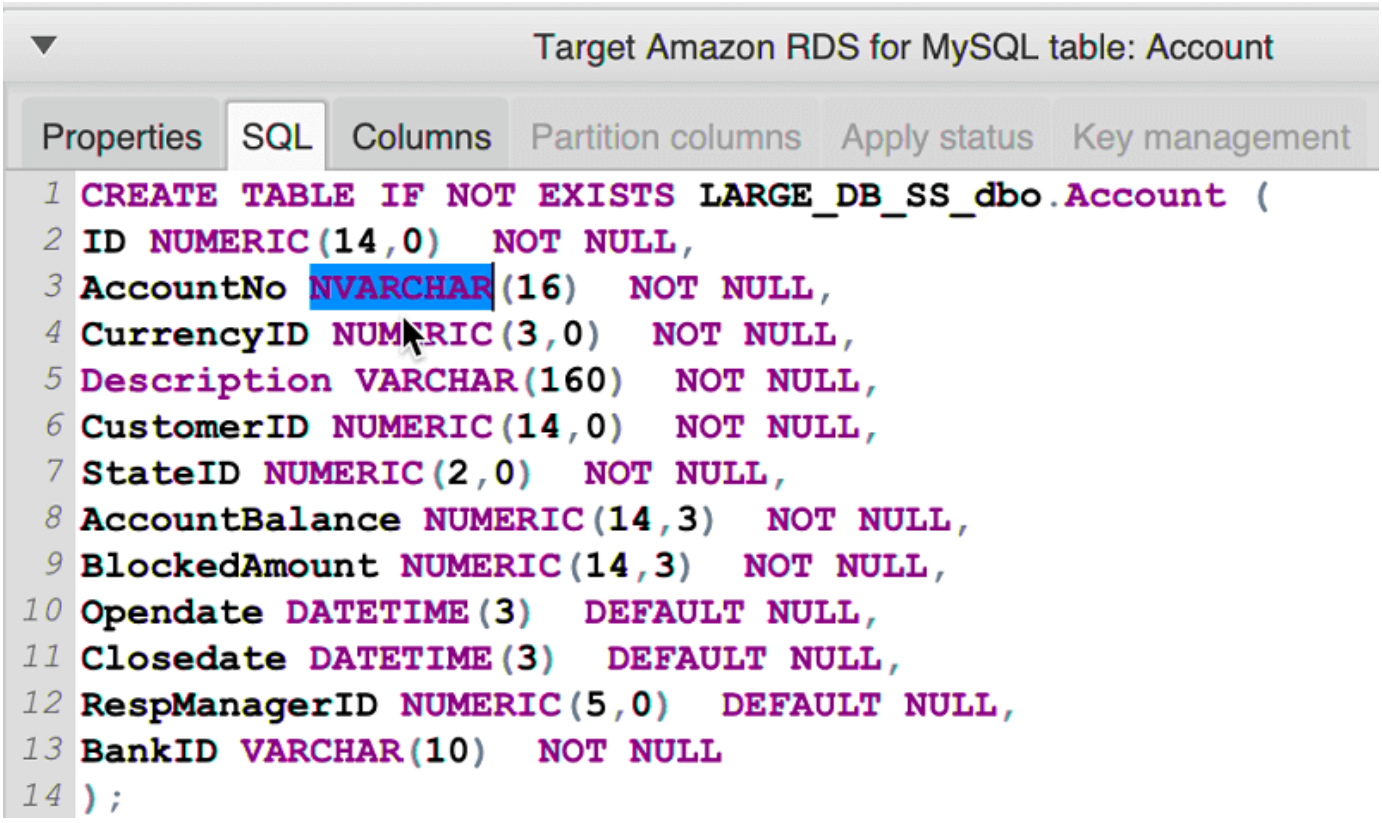
Cuando la AWS SCT genera un esquema convertido, no se aplica inmediatamente a la instancia de base de datos de destino. El esquema convertido se almacena en el entorno local hasta que esté listo para aplicarlo en la instancia de base de datos de destino. Para obtener más información, consulte Aplicación del esquema convertido.

Editar esquemas convertidos

Puede editar el esquema convertido y guardar los cambios como parte de su proyecto.

Para editar el esquema convertido

1. En el panel izquierdo que indica el esquema de la base de datos de origen, seleccione el elemento del esquema para el que quiera editar el esquema convertido.
2. En el panel inferior central donde se muestra el esquema convertido para el elemento seleccionado, haga clic en la pestaña SQL.
3. En el texto mostrado para la pestaña SQL, cambie el esquema según sea necesario. El esquema se guardará automáticamente con su proyecto cuando lo actualice.



Los cambios que realice al esquema convertido se almacenan con su proyecto a medida que realiza las actualizaciones. Si convierte un nuevo elemento de un esquema desde su base de datos de origen y ha realizado actualizaciones al esquema previamente convertido para ese elemento, dichas actualizaciones existentes se verán sustituidas por el elemento del esquema recientemente convertido, en función de su base de datos de origen.

Eliminar un esquema convertido

Hasta aplicar el esquema a su instancia de base de datos de destino, la AWS SCT solo almacena el esquema convertido localmente en su proyecto. Puede eliminar el esquema planificado de su proyecto si selecciona el nodo de vista en árbol para su instancia de base de datos y, a continuación, hace clic en Actualizar desde base de datos. Dado que no se ha escrito ningún esquema en su instancia de base de datos de destino, la actualización desde la base de datos eliminará los elementos del esquema planificado en su proyecto de AWS SCT para adaptarse a lo que exista en su instancia de base de datos de destino.

Administrar conversiones manuales en AWS SCT

El informe de evaluación incluye una lista de los elementos que no se pueden convertir automáticamente en el motor de base de datos de su instancia de base de datos de Amazon RDS de destino. Para cada elemento que no se puede convertir, hay un elemento de acción en la pestaña Action Items.

Puede responder a los elementos de acción del informe de evaluación de las siguientes formas:

- Modificar su esquema de base de datos de origen.
- Modificar su esquema de base de datos de destino.

Modificar su esquema de origen

Para algunos elementos, es posible que resulte más fácil modificar el esquema de base de datos en un esquema de base de datos de origen que lograr que se convierta automáticamente. En primer lugar, verifique que los nuevos cambios son compatibles con la arquitectura de su aplicación y, a continuación, actualice el esquema en la base de datos de origen. Por último, actualice su proyecto con la información de esquema actualizada. Después, puede convertir el esquema actualizado y generar de nuevo el informe de evaluación de la migración de la base de datos. Los elementos de acción ya no aparecen para los elementos que hayan cambiado en el esquema de origen.

La ventaja de este proceso es que el esquema actualizado está siempre disponible al actualizar desde la base de datos de origen.

Modificar su esquema de destino

Para algunos elementos, puede que sea más sencillo aplicar el esquema convertido a la base de datos de destino y, a continuación, agregar manualmente los elementos de esquema equivalentes a la base de datos de destino para los elementos que no se hayan podido convertir automáticamente. Puede escribir todo el esquema que se pueda convertir automáticamente en su instancia de base de datos de destino aplicando el esquema. Para obtener más información, consulte Guardar y aplicar un esquema convertido en AWS SCT.

El esquema que escrito en la instancia de base de datos de destino no contendrá los elementos que no se hayan podido convertir automáticamente. Después de aplicar esquemas a la instancia de base de datos de destino, puede crear manualmente esquemas en dicha instancia que sean equivalentes a los de la base de datos de origen. Los elementos de acción en el informe de evaluación de la migración de la base de datos contienen sugerencias sobre cómo crear un esquema equivalente.

> **Warning**
>
> Si crea manualmente esquemas en su instancia de base de datos de destino, guarde una copia de cualquier operación manual que haya realizado. Si aplica el esquema convertido desde su proyecto a su instancia de base de datos de destino de nuevo, se sobrescribirá el trabajo manual que haya realizado.

En algunos casos, no podrá crear un esquema equivalente en la instancia de base de datos de destino. Tal vez tenga que rediseñar una parte de la aplicación y de la base de datos para usar la funcionalidad disponible en el motor de base de datos para la instancia de base de datos de destino. En otros casos, puede omitir simplemente el esquema que no se pueda convertir automáticamente.

Actualizar un esquema convertido en AWS SCT

Puede actualizar el esquema de origen y el esquema de destino en su proyecto de AWS Schema Conversion Tool.

- Origen: si actualiza el esquema para su base de datos de origen, AWS SCT sustituye el esquema de su proyecto con el último esquema de su base de datos de origen. Con esta funcionalidad, puede actualizar su proyecto si se ha modificado el esquema de la base de datos de origen.
- Destino: si actualiza el esquema para su instancia de base de datos de Amazon RDS de destino, AWS SCT sustituye el esquema de su proyecto por el último esquema de su instancia de base de

datos de destino. Si no ha aplicado ningún esquema a su instancia de base de datos de destino, AWS SCT elimina el esquema convertido de su proyecto. A continuación, puede convertir el esquema de la base de datos de origen a una instancia de base de datos de destino limpia.

El esquema en su proyecto de la AWS SCT se actualiza seleccionando Actualizar desde base de datos, como se muestra a continuación.

> **Note**
>
> Al actualizar el esquema, AWS SCT carga los metadatos solo cuando son necesarios. Para cargar completamente todo el esquema de la base de datos, abra el menú contextual (clic secundario) del esquema y elija Cargar esquema. Por ejemplo, puede usar esta opción para cargar todos los metadatos de la base de datos de una sola vez y, a continuación, trabajar sin conexión.

Guardar y aplicar un esquema convertido en AWS SCT

Cuando la AWS Schema Conversion Tool genera un esquema convertido (tal y como se muestra en Convertir su esquema con AWS SCT), no aplica inmediatamente el esquema convertido a la instancia de base de datos de destino. Los esquemas convertidos se almacenan en el entorno local de su proyecto hasta que esté listo para aplicarlos en la instancia de base de datos de destino. Con esta funcionalidad, puede trabajar con los elementos de esquema que no se puedan convertir automáticamente en su motor de base de datos de destino. Para obtener más información sobre elementos que no se pueden convertir automáticamente, consulte Crear informes de evaluación de migración con AWS SCT.

Si lo desea, puede hacer que la herramienta guarde su esquema convertido en un archivo como script SQL antes de aplicar el esquema a su instancia de base de datos de destino. También puede hacer que la herramienta aplique el esquema convertido directamente a su instancia de base de datos de destino.

Guardar el esquema convertido en un archivo

Puede guardar los esquemas convertidos como scripts SQL en un archivo de texto. Al utilizar este enfoque, puede modificar los scripts SQL generados en la AWS SCT para abordar los elementos que no puedan convertirse automáticamente. A continuación, puede ejecutar sus scripts actualizados en

su instancia de base de datos de destino para aplicar el esquema convertido a la base de datos de destino.

Para guardar el esquema convertido como scripts de SQL

1. Elija su esquema y abra el menú contextual (clic secundario).
2. Elija Guardar como SQL.
3. Introduzca el nombre del archivo y seleccione Guardar.
4. Guarde el esquema convertido si opta por una de las opciones siguientes:

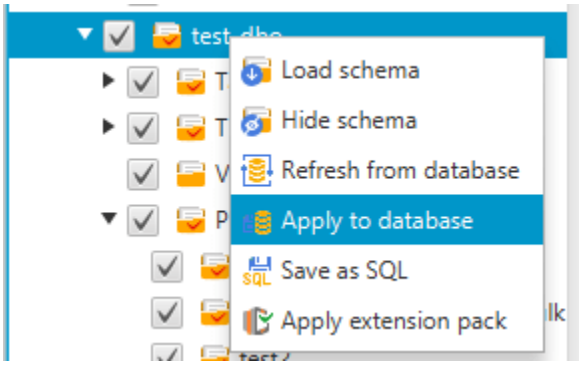
   - Archivo único
   - Archivo único por fase
   - Archivo único por instrucción

Para elegir el formato del script de SQL

1. En el menú Configuración, seleccione Configuración del proyecto.
2. Seleccione Guardar scripts.
3. En Proveedor, elija la plataforma de base de datos.
4. En Guardar scripts de SQL en, elija cómo desea guardar el script de esquema de base de datos.
5. Seleccione Aceptar para guardar la configuración.

Aplicación del esquema convertido

Cuando esté listo para aplicar el esquema convertido a la instancia de base de datos de Amazon RDS de destino, elija el elemento del esquema en el panel derecho del proyecto. Abra el menú contextual (clic con el botón secundario) del elemento del esquema y seleccione Apply to database, como se muestra a continuación.



Esquema del paquete de extensión

La primera vez que aplique el esquema convertido a su instancia de base de datos de destino, la AWS SCT agrega un esquema adicional a su instancia de base de datos de destino. Este esquema implementa las funciones del esquema de la base de datos que son necesarias a la hora de escribir el esquema convertido en la instancia de base de datos de destino. El esquema se denomina esquema de paquete de extensión.

No modifique el esquema del paquete de extensión, ya que podría encontrarse resultados imprevistos en el esquema convertido creado en la instancia de base de datos de destino. Cuando el esquema haya migrado totalmente a su instancia de base de datos de destino y ya no necesite la AWS SCT, puede eliminar el esquema del paquete de extensión.

El esquema de paquete de extensión se denomina en función de su base de datos de origen, de la siguiente manera:

- IBM Db2 LUW: `aws_db2_ext`
- Microsoft SQL Server: `aws_sqlserver_ext`
- MySQL: `aws_mysql_ext`
- Oracle: `aws_oracle_ext`
- PostgreSQL: `aws_postgresql_ext`
- SAP ASE: `aws_sapase_ext`

Para obtener más información, consulte Uso de las AWS Lambda funciones del paquete de AWS SCT extensión .

Comparación de esquemas de bases de datos

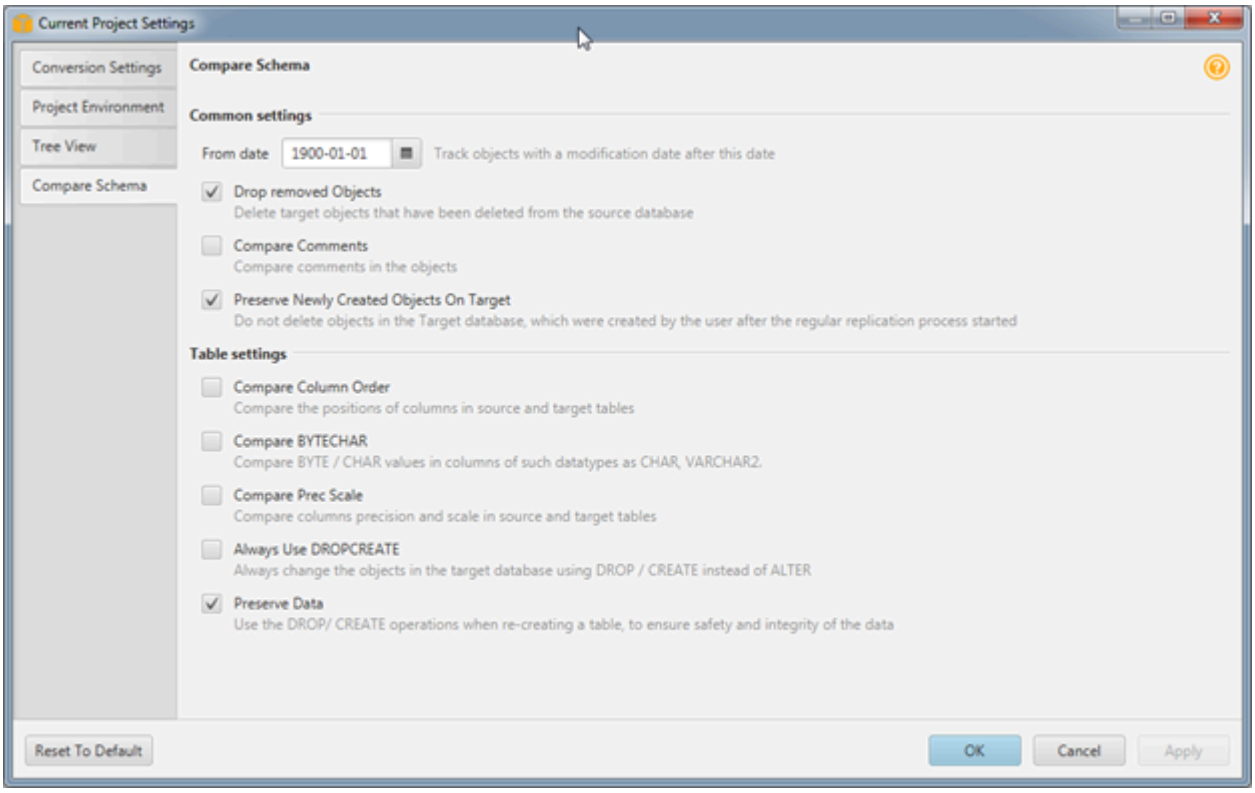
Si ha realizado cambios en su esquema de origen o de destino después de la migración, puede comparar los dos esquemas de base de datos mediante AWS SCT. Puede comparar esquemas de versiones iguales o anteriores al esquema de origen.

Se admiten las siguientes comparaciones de esquemas:

- De Oracle a Oracle, versiones 12.1.0.2.0, 11.1.0.7.0, 11.2.0.1.0, 10
- De SQL Server a SQL Server, versiones 2016, 2014, 2012, 2008 RD2, 2008

- De PostgreSQL a PostgreSQL y Aurora PostgreSQL-Compatible Edition, versiones 9.6, 9.5.9, 9.5.4
- De MySQL a MySQL, versiones 5.6.36, 5.7.17, 5.5

La configuración de la comparación de esquemas se especifica en la pestaña Compare Schema de la página Project Settings.

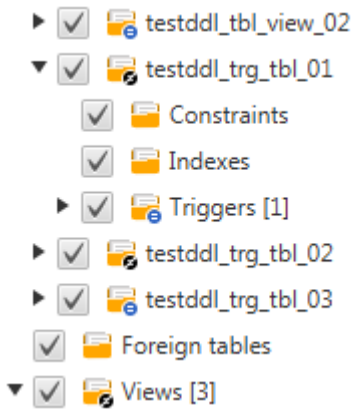


Para comparar esquemas, el usuario selecciona los esquemas y AWS SCT indica los objetos que difieren entre los dos esquemas y los que no.

Para comparar dos esquemas

1. Abra un proyecto de AWS SCT existente o cree un proyecto y conéctese a los puntos de enlace de origen y destino.
2. Elija el esquema que desea comparar.
3. Abra el menú contextual (clic secundario) y elija Comparar esquema.

AWS SCT indica los objetos que son diferentes entre los dos esquemas agregando un círculo negro al icono del objeto.



Puede aplicar los resultados de la comparación de esquemas a un único objeto, a una única categoría de objetos o a todo el esquema. Seleccione la casilla situada junto a la categoría, objeto o esquema al que desee aplicar los resultados.

Buscar objetos transformados relacionados

Después de la conversión de un esquema, en algunos casos AWS SCT podría haber creado varios objetos para un objeto de esquema en la base de datos de origen. Por ejemplo, cuando se realiza una conversión de Oracle a PostgreSQL, AWS SCT toma cada disparador de Oracle y lo transforma en un disparador y una función disparadora en el destino de PostgreSQL. Además, cuando AWS SCT convierte una función o procedimiento de un paquete de Oracle en PostgreSQL, crea una función equivalente y una función INIT que debe ejecutarse como un bloque init para que el procedimiento o la función puedan ejecutarse.

El siguiente procedimiento le permite ver todos los objetos relacionados que se han creado después de una conversión del esquema.

Para ver los objetos relacionados que se crearon durante una conversión de esquema

1. Tras la conversión del esquema, elija el objeto convertido en la vista de árbol de destino.
2. Elija la pestaña Related Converted Objects.
3. Consulte la lista de objetos de destino relacionados.

Convertir esquemas de almacenamiento de datos a Amazon Redshift mediante AWS SCT

AWS Schema Conversion Tool (AWS SCT) automatiza gran parte del proceso de conversión de su esquema de almacenamiento de datos a un esquema de base de datos de Amazon Redshift. Debido a que los motores de base de datos de origen y de destino pueden tener diferentes características y capacidades, AWS SCT intenta crear un esquema equivalente en su base de datos de destino siempre que sea posible. Si no es posible la conversión directa, AWS SCT proporciona un informe de evaluación con una lista de posibles acciones que puede realizar. Con AWS SCT, puede administrar claves, mapear tipos de datos y objetos, y crear conversiones manuales.

AWS SCT puede convertir los siguientes esquemas de almacenamiento de datos a Amazon Redshift.

- Amazon Redshift
- Azure Synapse Analytics (versión 10)
- BigQuery
- Greenplum Database (versión 4.3)
- Microsoft SQL Server (versión 2008 y posteriores)
- Netezza (versión 7.0.3 y posteriores)
- Oracle (versión 10.2 y posteriores)
- Snowflake (versión 3)
- Teradata (versión 13 y posteriores)
- Vertica (versión 7.2 y posteriores)

Si información sobre cómo convertir un esquema de base de datos de procesamiento de transacciones online (OLTP), consulte Conversión de esquemas de base de datos mediante AWS SCT.

Para convertir un esquema de almacenamiento de datos, siga estos pasos:

1. Especifique la estrategia de optimización y las reglas, y especifique las reglas de migración que desea que utilice AWS SCT. Puede configurar reglas que cambien el tipo de datos de las columnas, mover objetos de un esquema a otro y cambiar los nombres de los objetos.

Puede especificar la optimización y las reglas de migración en Configuración. Para obtener más información acerca de estrategias de optimización, consulte Seleccionar estrategias de optimización y reglas de uso para usar con AWS SCT. Para obtener más información sobre la reglas de migración, consulte Crear reglas de migración en AWS SCT

2. Proporcione estadísticas de su almacenamiento de datos de origen para que AWS SCT pueda optimizar cómo se convierte su almacenamiento de datos. Puede recopilar estadísticas directamente desde la base de datos o cargar un archivo de estadísticas existente. Para obtener más información acerca de cómo proporcionar estadísticas de almacenamiento de datos, consulte Recopilar o cargar estadísticas para AWS SCT.
3. Cree un informe de evaluación de la migración de la base de datos en el que se detallen los elementos del esquema que no se pueden convertir automáticamente. Puede utilizar este informe para identificar dónde necesita crear manualmente un esquema en su base de datos de destino que sea compatible con su base de datos de origen. Para obtener más información sobre el informe de evaluación, consulte Crear informes de evaluación de migración con AWS SCT.
4. Convertir el esquema: la AWS SCT crea una versión local del esquema convertido para que lo revise, pero no se aplica a su base de datos de destino hasta que esté listo. Para obtener más información acerca de la conversión, consulte Convertir su esquema mediante AWS SCT.
5. Tras convertir el esquema, puede administrar y editar sus claves. La administración de claves es la esencia de la conversión de un almacenamiento de datos. Para obtener más información acerca de la administración de claves derivadas, consulte Administrar y personalizar claves en AWS SCT.
6. Si tiene elementos del esquema que no se pueden convertir automáticamente, dispone de dos opciones: actualizar el esquema de origen y, a continuación, volver a convertirlo, o bien crear elementos de esquema equivalentes en su base de datos de destino. Para obtener más información acerca de cómo convertir manualmente elementos de esquema, consulte Administrar conversiones manuales en AWS SCT. Para obtener más información acerca de cómo actualizar su esquema de origen, consulte Actualizar un esquema convertido en AWS SCT.
7. Cuando esté listo, puede aplicar el esquema convertido a la base de datos de destino. Para obtener más información acerca de cómo guardar y aplicar el esquema convertido, consulte Guardar y aplicar un esquema convertido en AWS SCT.

Permisos para Amazon Redshift como destino

A continuación se enumeran los permisos necesarios para Amazon Redshift como destino:

- CREATE ON DATABASE: permite crear nuevos esquemas en la base de datos.

- CREATE ON SCHEME: permite crear objetos en el esquema de la base de datos.
- GRANT USAGE ON LANGUAGE: permite crear nuevas funciones y procedimientos en la base de datos.
- GRANT SELECT ON ALL TABLES IN SCHEME pg_catalog: proporciona al usuario información del sistema sobre el clúster de Amazon Redshift.
- GRANT SELECT ON pg_class_info: proporciona al usuario información sobre el estilo de distribución de las tablas.

Puede usar el siguiente ejemplo de código para crear un usuario de base de datos y conceder los permisos correspondientes.

```
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;
```

En el ejemplo anterior, sustituya *user_name* por el nombre de su usuario. A continuación, sustituya *db_name* por el nombre de la base de datos de Amazon Redshift de destino. A continuación, sustituya *schema_name* por el nombre del esquema de Amazon Redshift. Repita la operación `GRANT CREATE ON SCHEMA` para cada esquema de destino en el que vaya a aplicar el código convertido o a migrar los datos. Por último, sustituya *your_password* por una contraseña segura.

Puede aplicar un paquete de extensión a la base de datos de Amazon Redshift de destino. Un paquete de extensión es un módulo complementario que simula funciones de la base de datos de origen que son necesarias para convertir objetos a Amazon Redshift. Para obtener más información, consulte Uso de paquetes AWS SCT de extensión.

Para esta operación, AWS SCT necesita permiso para acceder al bucket de Amazon S3 en su nombre. Para conceder este permiso, cree un usuario de AWS Identity and Access Management (IAM) con la siguiente política.

```
{
   "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":""
      }
   ]
}
```

Seleccionar estrategias de optimización y reglas de uso para usar con AWS SCT

Para optimizar el modo en que AWS Schema Conversion Tool convierte su esquema de almacenamiento de datos, puede elegir las estrategias y reglas que desea que utilice la herramienta. Tras convertir su esquema y revisar las claves propuestas, puede ajustar sus reglas o cambiar su estrategia para obtener los resultados que quiera.

Para seleccionar sus estrategias y reglas de optimización

1. Seleccione Configuración y después Configuración del proyecto. Aparecerá el cuadro de diálogo Configuración del proyecto actual.
2. En el panel izquierdo, seleccione Estrategias de optimización. Las estrategias de optimización aparecerán en el panel derecho, con los valores predeterminados seleccionados.
3. En Sector de estrategia, seleccione la estrategia de optimización que quiera emplear. Puede elegir entre las siguientes opciones:

- Usar metadatos y omitir información estadística: con esta estrategia, solo se usa la información de los metadatos para decisiones sobre optimización. Por ejemplo, si hay más de un índice en una tabla de origen, se utiliza el orden de clasificación de la base de datos de origen y el primer índice se convierte en una clave de distribución.

- Omitir metadatos y usar información estadística: con esta estrategia, las decisiones de optimización se derivan solo de la información estadística. Esta estrategia se aplica únicamente a las tablas y columnas para las que se proporcionan estadísticas. Para obtener más información, consulte Recopilar o cargar estadísticas para AWS SCT.

- Usar metadatos y usar información estadística: con esta estrategia, tanto los metadatos como las estadísticas se usan para las decisiones de optimización.

4. Tras seleccionar su estrategia de optimización, podrá determinar qué reglas quiere usar. Puede elegir entre las siguientes opciones:

   - Seleccionar clave de distribución y claves de ordenación con metadatos
   - Seleccionar tabla de hechos y una dimensión apropiada para la intercalación
   - Analizar la cardinalidad de las columnas de índices
   - Determinar las tablas y columnas más utilizadas desde la tabla de registro de consultas

   Para cada regla, puede introducir un peso para la clave de clasificación y un peso para la clave de distribución. AWS SCT utiliza los pesos que elija cuando convierte su esquema. Después, cuando analice las claves sugeridas, si no está satisfecho con los resultados, puede volver aquí y cambiar la configuración. Para obtener más información, consulte Administrar y personalizar claves en AWS SCT.

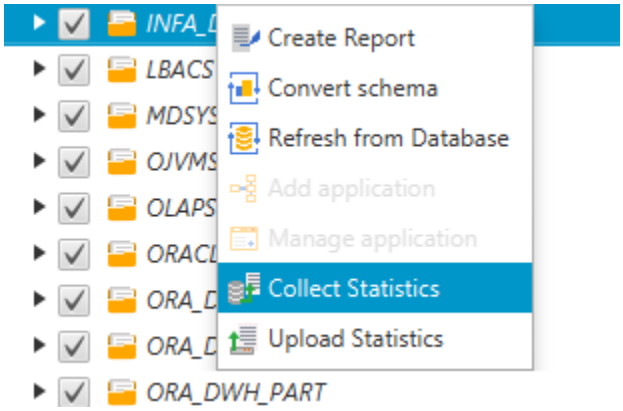
Recopilar o cargar estadísticas para AWS SCT

Para optimizar el modo en que AWS Schema Conversion Tool convierte su esquema de almacenamiento de datos puede proporcionar estadísticas desde su base de datos de origen que

la herramienta podrá utilizar. Puede recopilar estadísticas directamente desde la base de datos o cargar un archivo de estadísticas existente.

Para facilitar y revisar las estadísticas

1. Abra su proyecto y conéctese a la base de datos de origen.
2. Seleccione un objeto de esquema desde el panel izquierdo de su proyecto y abra el menú contextual (clic secundario) para el objeto. Seleccione Recopilar estadísticas o Cargar estadísticas como se muestra a continuación.



3. Seleccione un objeto de esquema del panel izquierdo de su proyecto y, a continuación, haga clic en la pestaña Estadísticas. Puede revisar las estadísticas para el objeto.

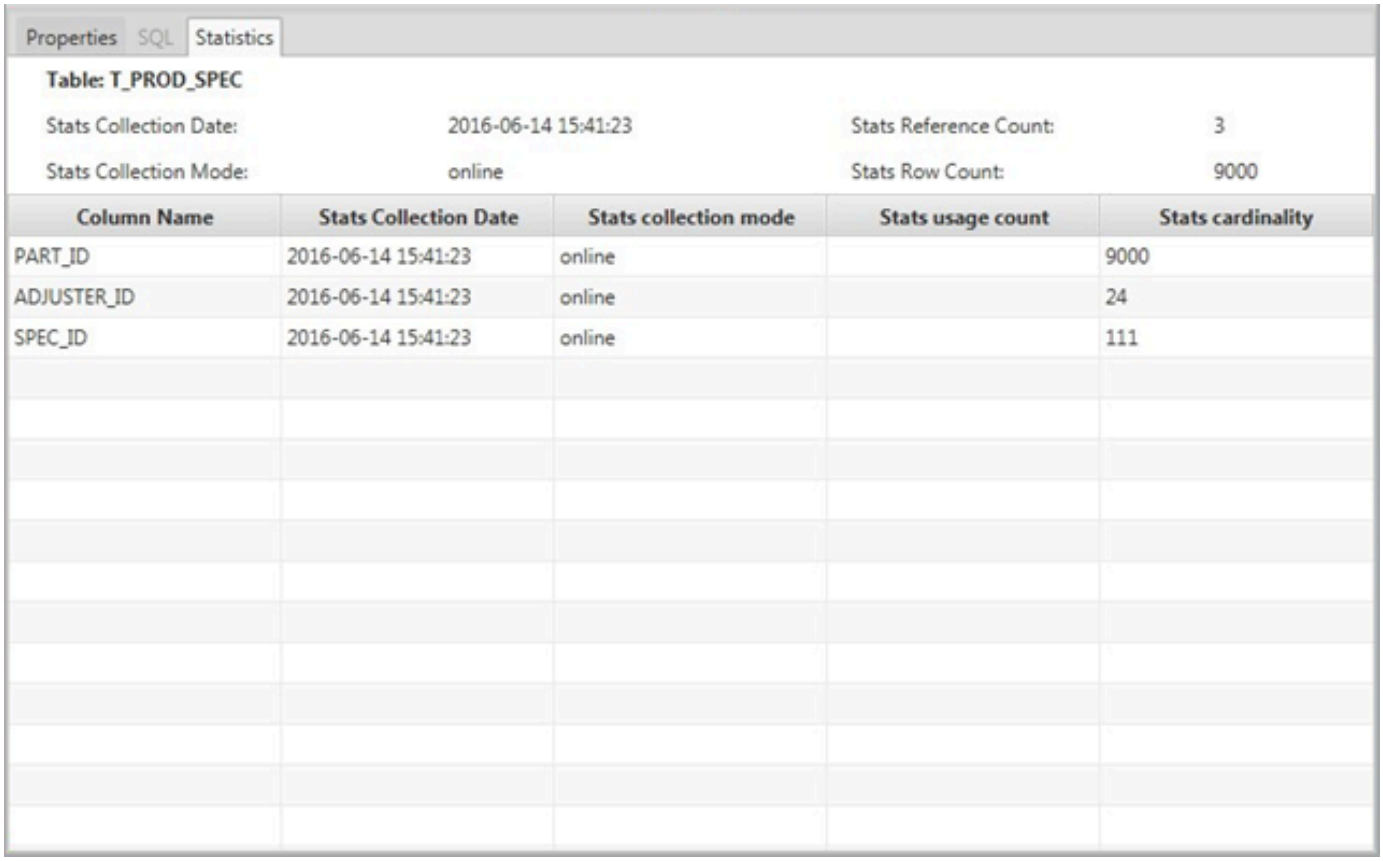

Properties SQL Statistics

Table: T_PROD_SPEC

Stats Collection Date: 2016-06-14 15:41:23 Stats Reference Count: 3

Stats Collection Mode: online Stats Row Count: 9000

Column Name	Stats Collection Date	Stats collection mode	Stats usage count	Stats cardinality
PART_ID	2016-06-14 15:41:23	online		9000
ADJUSTER_ID	2016-06-14 15:41:23	online		24
SPEC_ID	2016-06-14 15:41:23	online		111

Después, cuando analice las claves sugeridas, si no está satisfecho con los resultados, puede recopilar estadísticas adicionales y repetir el procedimiento. Para obtener más información, consulte Administrar y personalizar claves en AWS SCT.

Crear reglas de migración en AWS SCT

Antes de convertir el esquema con AWS SCT, puede configurar las reglas de migración. Las reglas de migración pueden hacer cosas como cambiar el tipo de datos de las columnas, mover objetos de un esquema a otro y cambiar los nombres de los objetos. Por ejemplo, suponga que tiene un conjunto de tablas en el esquema de origen denominado `test_TABLE_NAME`. Puede configurar una regla que cambie el prefijo `test_` por el prefijo `demo_` del esquema de destino.

> **Note**
>
> Puede crear reglas de migración solo para motores de bases de datos de origen y destino diferentes.

Puede crear reglas de migración que realicen las siguientes tareas:

- Agregar, eliminar o reemplazar un prefijo
- Agregar, eliminar o reemplazar un sufijo
- Cambiar intercalación de columnas
- Cambiar el tipo de datos
- Cambiar la longitud de `char` `varcharnvarchar`, y los tipos de datos `string`
- Mover objetos
- Cambiar el nombre de los objetos

Puede crear reglas de migración para los siguientes objetos:

- Base de datos
- Esquema
- Tabla
- Columna

Crear reglas de migración

Puede crear reglas de migración y guardarlas como parte de su proyecto. Con el proyecto abierto, utilice el siguiente procedimiento para crear reglas de migración.

Para crear reglas de migración

1. En el menú Ver, seleccione Vista principal.
2. En Asignaciones de servidores, elija un par de servidores de origen y de destino.
3. Elija Regla de migración nueva. Aparecerá el cuadro de diálogo Reglas de transformación.
4. Seleccione Agregar regla nueva. Se agregará una nueva fila a la lista de reglas.
5. Configure la regla:

    a. En Nombre, ingrese un nombre para la regla.

    b. En Para, seleccione el tipo de objeto al que se aplicará la regla.

    c. En where, introduzca un filtro que se aplicará a los objetos antes de aplicar la regla de migración. La cláusula where se evalúa mediante una cláusula like. Puede especificar un

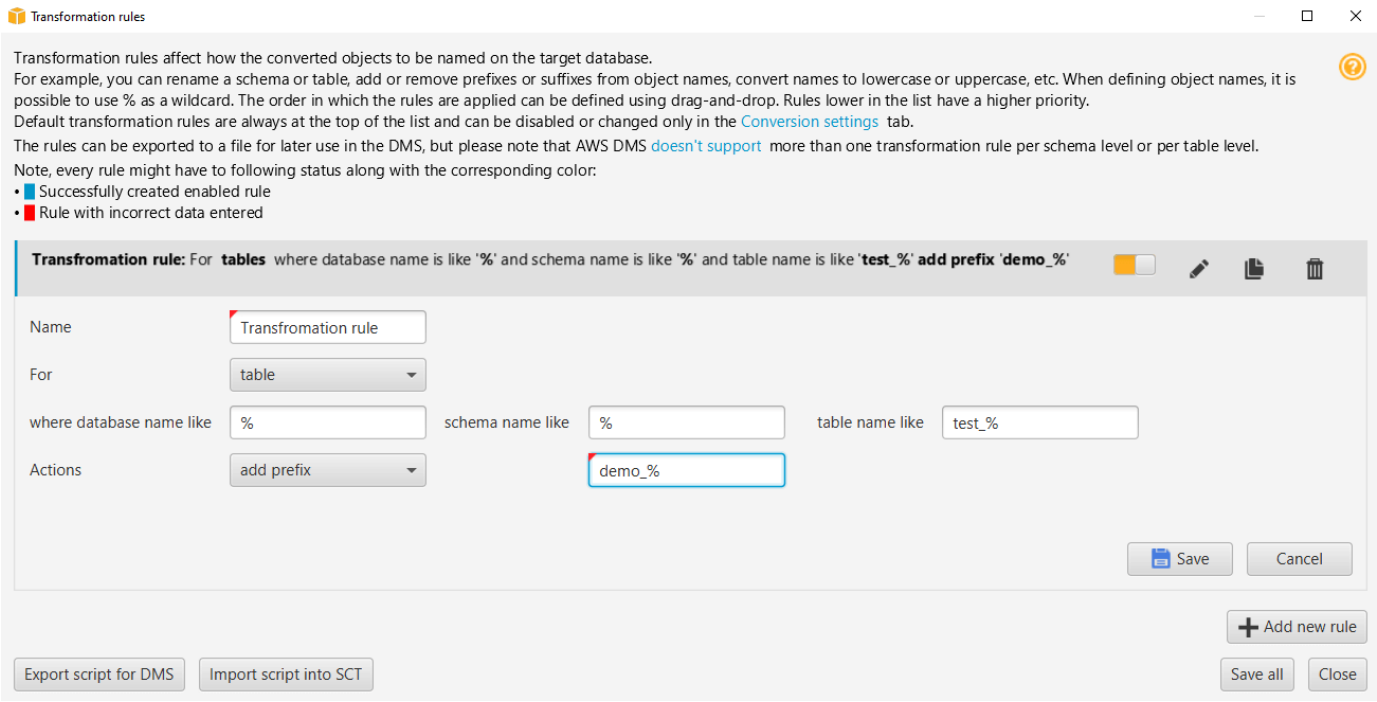
nombre exacto para seleccionar un objeto o introducir un patrón para seleccionar varios objetos.

Los campos disponibles para la cláusula where son diferentes en función del tipo de objeto. Por ejemplo, si el tipo de objeto es esquema, solo habrá un campo disponible, para el nombre del esquema.

d. En Acciones, seleccione el tipo de regla de migración que quiera crear.

e. En función del tipo de regla, introduzca uno o dos valores adicionales. Por ejemplo, para cambiar el nombre de un objeto, introduzca el nuevo nombre del objeto. Para sustituir un prefijo, introduzca el prefijo anterior y el nuevo prefijo.

6. Tras haber configurado su regla de migración, seleccione Guardar para guardar su regla. También puede seleccionar Cancelar para cancelar los cambios.



7. Cuando haya acabado de agregar, editar y eliminar reglas, seleccione Guardar todo para guardar todos los cambios.

8. Seleccione Cerrar para cerrar el cuadro de diálogo Reglas de transformación.

Puede utilizar el icono de alternar para desactivar una regla de migración sin eliminarla. Puede utilizar el icono de copia para duplicar una regla de migración existente. Puede utilizar el icono del lapicero para editar una regla de migración existente. Puede utilizar el icono de eliminar para borrar

una regla de migración existente. Para guardar los cambios que realice en sus reglas de migración, seleccione Guardar todo.

Exportar reglas de migración

Si utiliza AWS Database Migration Service (AWS DMS) para migrar sus datos de la base de datos de origen a la de destino, puede proporcionar información sobre sus reglas de migración a AWS DMS. Para obtener más información sobre las tareas, consulte Trabajar con tareas de replicación de AWS Database Migration Service.

Para exportar reglas de migración

1. En AWS Schema Conversion Tool, seleccione Vista de asignaciones en el menú Vista.
2. En Reglas de migración, elija una regla de migración y, a continuación, elija Modificar regla de migración.
3. Elija Exportar script para AWS DMS.
4. Seleccione la ubicación en la que desee guardar el script y después seleccione Guardar. Su reglas de migración se guardan como un script JSON que AWS DMS puede consumir.

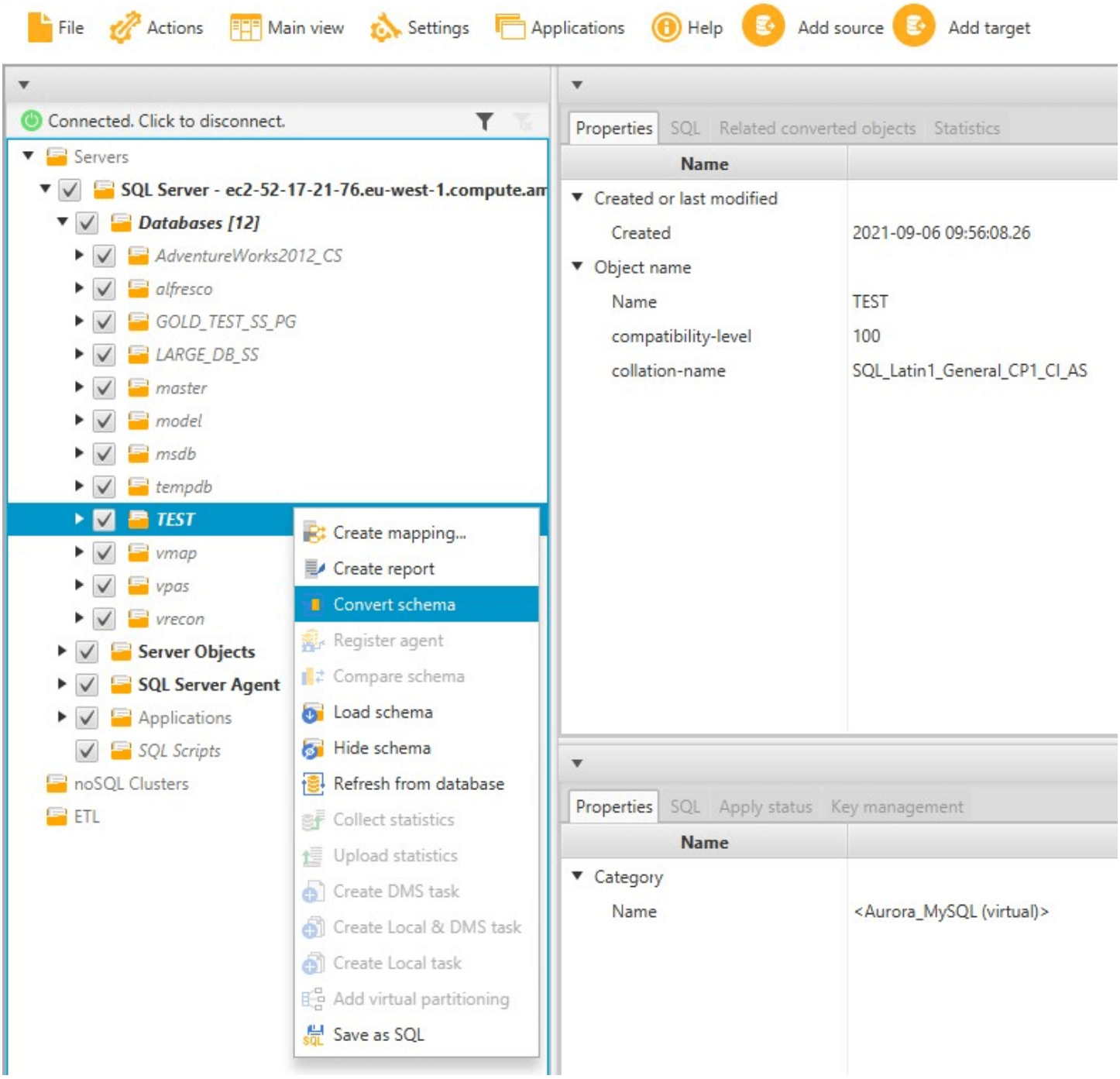
Convertir su esquema mediante AWS SCT

Tras haber conectado su proyecto tanto a la base de datos de origen como a la base de datos de destino, su proyecto de AWS Schema Conversion Tool mostrará el esquema de su base de datos de origen en el panel izquierdo. El esquema se presentan en un formato de vista de árbol y cada nodo del árbol se carga progresivamente. Al seleccionar un nodo en la vista de árbol, AWS SCT solicita la información del esquema de la base de datos de origen en ese momento.

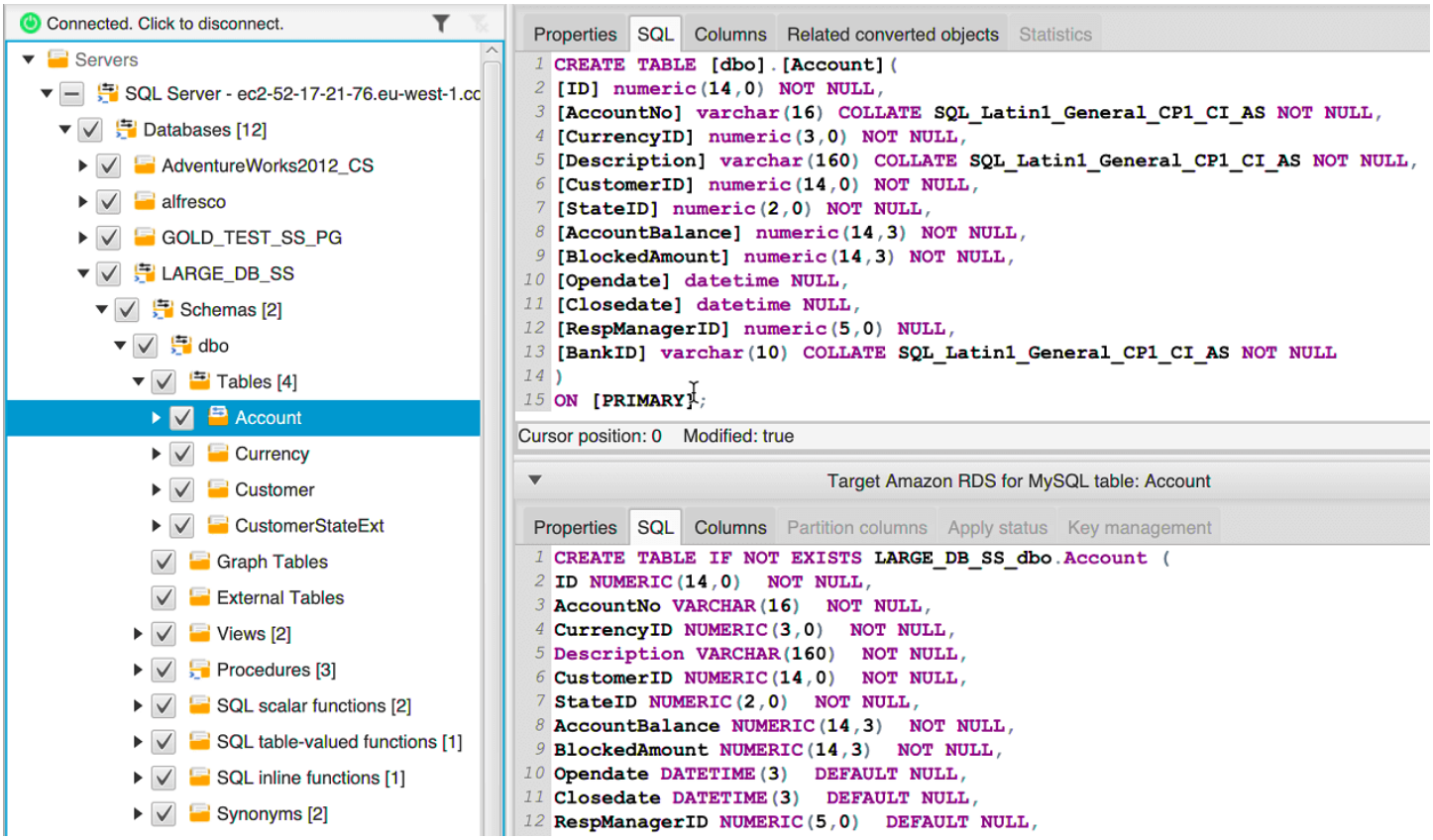
Puede seleccionar elementos del esquema en su base de datos de origen y, después, convertir el esquema en un esquema equivalente para el motor de la base de datos de su base de datos de destino. Puede seleccionar cualquier elemento del esquema en su base de datos de origen para convertirlo. Si el elemento del esquema que seleccione depende de un elemento principal, AWS SCT generará también el esquema para el elemento principal. Por ejemplo, si selecciona una columna de una tabla para convertirla, AWS SCT generará el esquema para la columna, la tabla en la que está la columna y la base de datos en la que está la tabla.

Convertir esquemas

Para convertir un esquema de la base de datos de origen, active la casilla de verificación del nombre del esquema que desee convertir. A continuación, seleccione este esquema en el panel izquierdo del proyecto. AWS SCT resalta el nombre del esquema en azul. Abra el menú contextual (clic secundario) para el esquema y seleccione Convertir esquemas, como se muestra a continuación.



Una vez que haya convertido el esquema de su base de datos de origen, puede elegir los elementos del esquema en el panel izquierdo de su proyecto y ver el esquema convertido en los paneles centrales de su proyecto. El panel inferior central muestra las propiedades y el comando SQL para crear el esquema convertido, como se muestra a continuación.



Tras haber convertido el esquema, podrá guardar el proyecto. La información del esquema de su base de datos de origen se guarda con su proyecto. Esta funcionalidad supone que puede trabajar sin conexión sin estar conectado a su base de datos de origen. AWS SCT se conecta a su base de datos de origen para actualizar el esquema en su proyecto si selecciona Refresh from Database (Actualizar desde base de datos) para su base de datos de origen. Para obtener más información, consulte Actualizar un esquema convertido en AWS SCT.

Puede crear un informe de evaluación de la migración de la base de datos para los elementos que no se puedan convertir automáticamente. El informe de evaluación es útil para identificar y solucionar los elementos del esquema que no se puedan convertir automáticamente. Para obtener más información, consulte Crear informes de evaluación de migración con AWS SCT.

Cuando la AWS SCT genera un esquema convertido, no se aplica inmediatamente a la base de datos de destino. El esquema convertido se almacena en el entorno local hasta que esté listo para

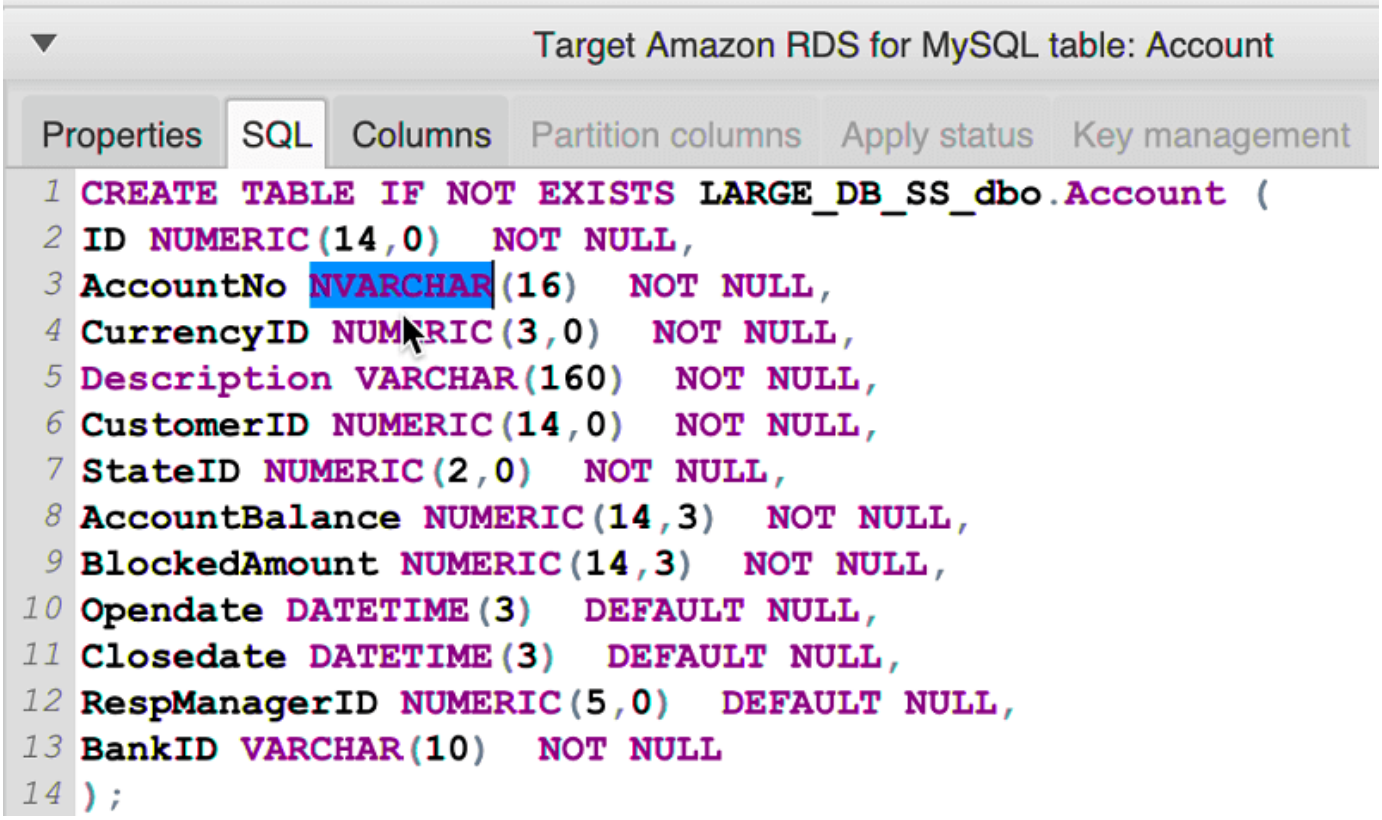
aplicarlo en la base de datos de destino. Para obtener más información, consulte Aplicar el esquema convertido.

Editar el esquema convertido

Puede editar el esquema convertido y guardar los cambios como parte de su proyecto.

Para editar el esquema convertido

1. En el panel izquierdo que indica el esquema de la base de datos de origen, seleccione el elemento del esquema para el que quiera editar el esquema convertido.
2. En el panel inferior central donde se muestra el esquema convertido para el elemento seleccionado, haga clic en la pestaña SQL.
3. En el texto mostrado para la pestaña SQL, cambie el esquema según sea necesario. El esquema se guardará automáticamente con su proyecto cuando lo actualice.



Los cambios que realice al esquema convertido se almacenan con su proyecto a medida que realiza las actualizaciones. Si convierte un nuevo elemento de un esquema desde su base de datos de origen y ha realizado actualizaciones al esquema previamente convertido para ese elemento,

dichas actualizaciones existentes se verán sustituidas por el elemento del esquema recientemente convertido, en función de su base de datos de origen.

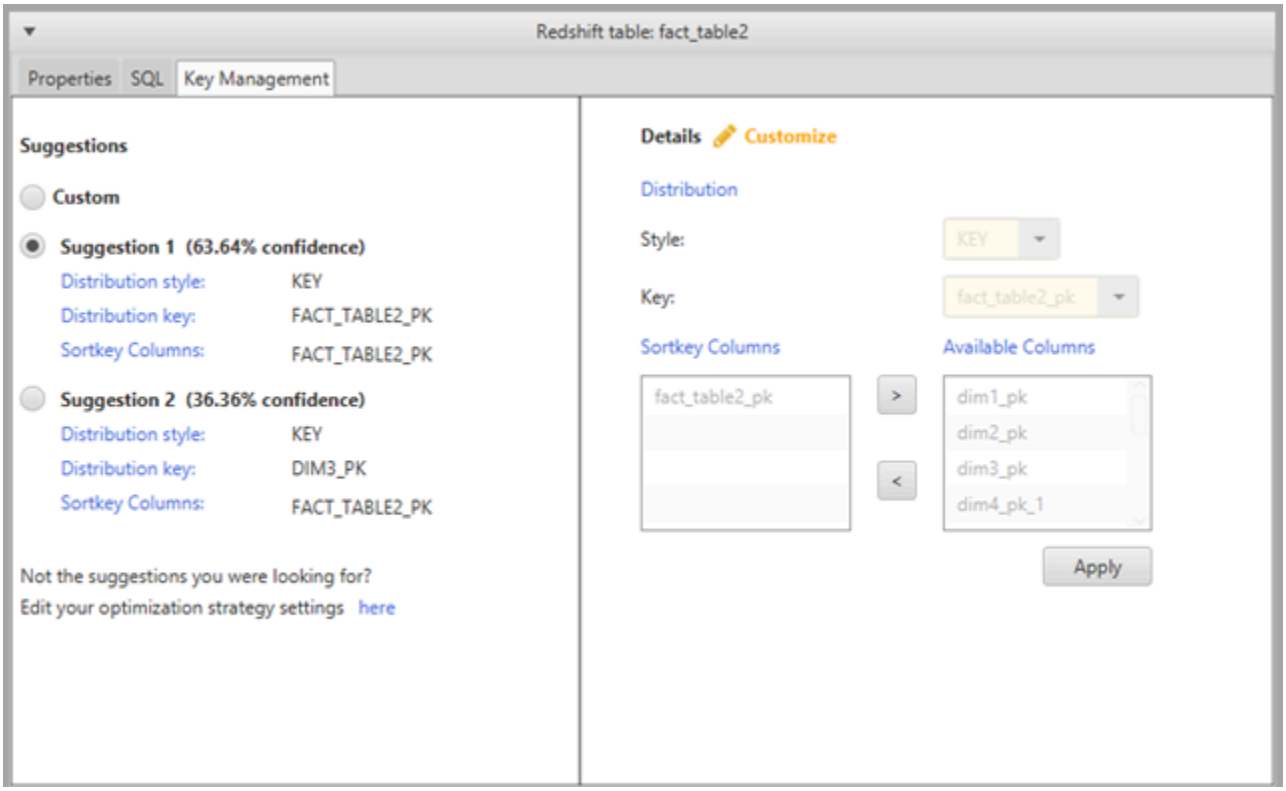
Eliminar un esquema convertido

Hasta aplicar el esquema a su base de datos de destino, la AWS SCT solo almacena el esquema convertido localmente en su proyecto. Puede eliminar el esquema planificado de su proyecto si selecciona el nodo de vista en árbol para su base de datos de destino y, a continuación, hace clic en Actualizar desde la base de datos. Dado que no se ha escrito ningún esquema en su base de datos de destino, la actualización desde la base de datos eliminará los elementos del esquema planificado en su proyecto de la AWS SCT para adaptarse a lo que exista en su base de datos de destino.

Administrar y personalizar claves en AWS SCT

Tras convertir el esquema con la AWS Schema Conversion Tool, puede administrar y editar sus claves. La administración de claves es la esencia de la conversión de un almacenamiento de datos.

Para administrar claves, seleccione una tabla en su base de datos de destino y a continuación seleccione la pestaña Administración de claves, como se muestra a continuación.



El panel izquierdo contiene sugerencias claves, e incluye la puntuación de fiabilidad para cada sugerencia. Puede seleccionar una de las sugerencias, o personalizar la clave editándola en el panel derecho.

Si las opciones de la clave no tienen el aspecto esperado, puede editar sus estrategias de optimización y volver a intentar la conversión. Para obtener más información, consulte Seleccionar estrategias de optimización y reglas de uso para usar con AWS SCT.

Temas relacionados

- Elegir la clave de clasificación recomendada
- Elegir el modo de distribución recomendado

Crear y usar el informe de evaluación en AWS SCT

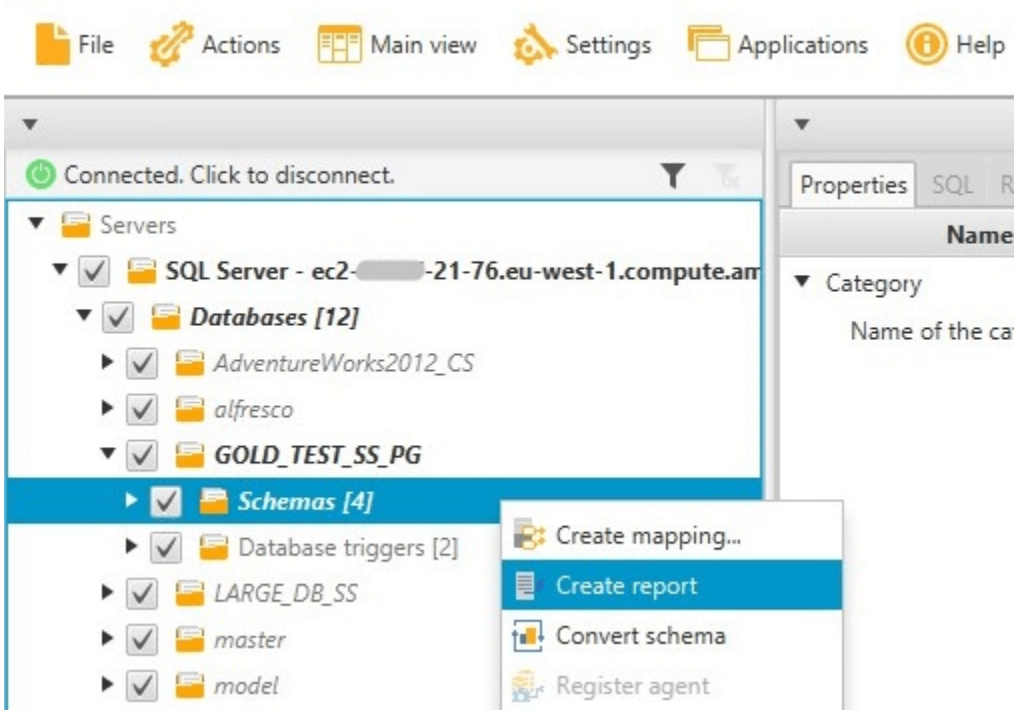
AWS Schema Conversion Tool crea un informe de evaluación de la migración de la base de datos para ayudarle a convertir su esquema. El informe de evaluación de la migración de la base de datos proporciona información importante sobre la conversión del esquema de la base de datos de origen a la base de datos de destino. El informe resume todas las tareas de conversión del esquema y detalla los elementos de acción del esquema que no se hayan podido convertir al motor de la base de datos de destino. El informe incluye también estimaciones de la cantidad de esfuerzo necesario para escribir el código equivalente en su base de datos de destino que no se pudo convertir automáticamente.

Crear un informe de evaluación de la migración de la base de datos

Utilice el siguiente procedimiento para crear un informe de evaluación de la migración de la base de datos.

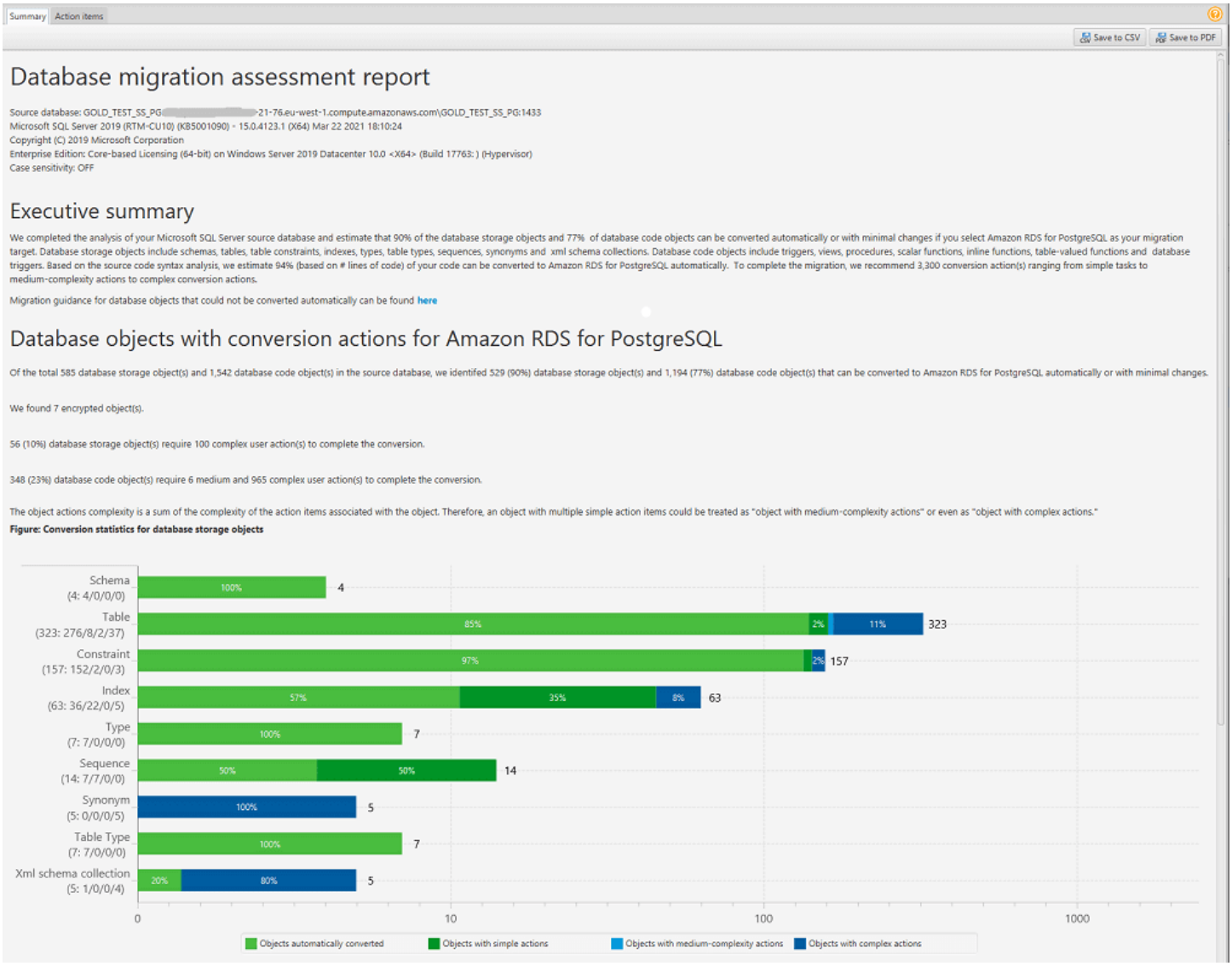
Para crear un informe de evaluación de la migración de la base de datos

1. En el panel de la izquierda que muestra el esquema de la base de datos de origen, elija el objeto del esquema para el que desea crear un informe de evaluación.
2. Abra el menú contextual (clic secundario) para el objeto y seleccione Crear informe.



Resumen de un informe de evaluación

Tras crear un informe de evaluación, se abrirá la vista del informe de evaluación, mostrando la pestaña Resumen. En la pestaña Resumen se muestra la información resumida del informe de evaluación de la migración de la base de datos. Muestra los elementos que se hayan convertido automáticamente y los elementos que no se hayan convertido automáticamente.



Para los elementos del esquema que no se puedan convertir automáticamente al motor de base de datos de destino, el resumen incluye una estimación del esfuerzo necesario para crear elementos del esquema en su instancia de base de datos de destino que sean equivalentes a los de su base de datos de origen.

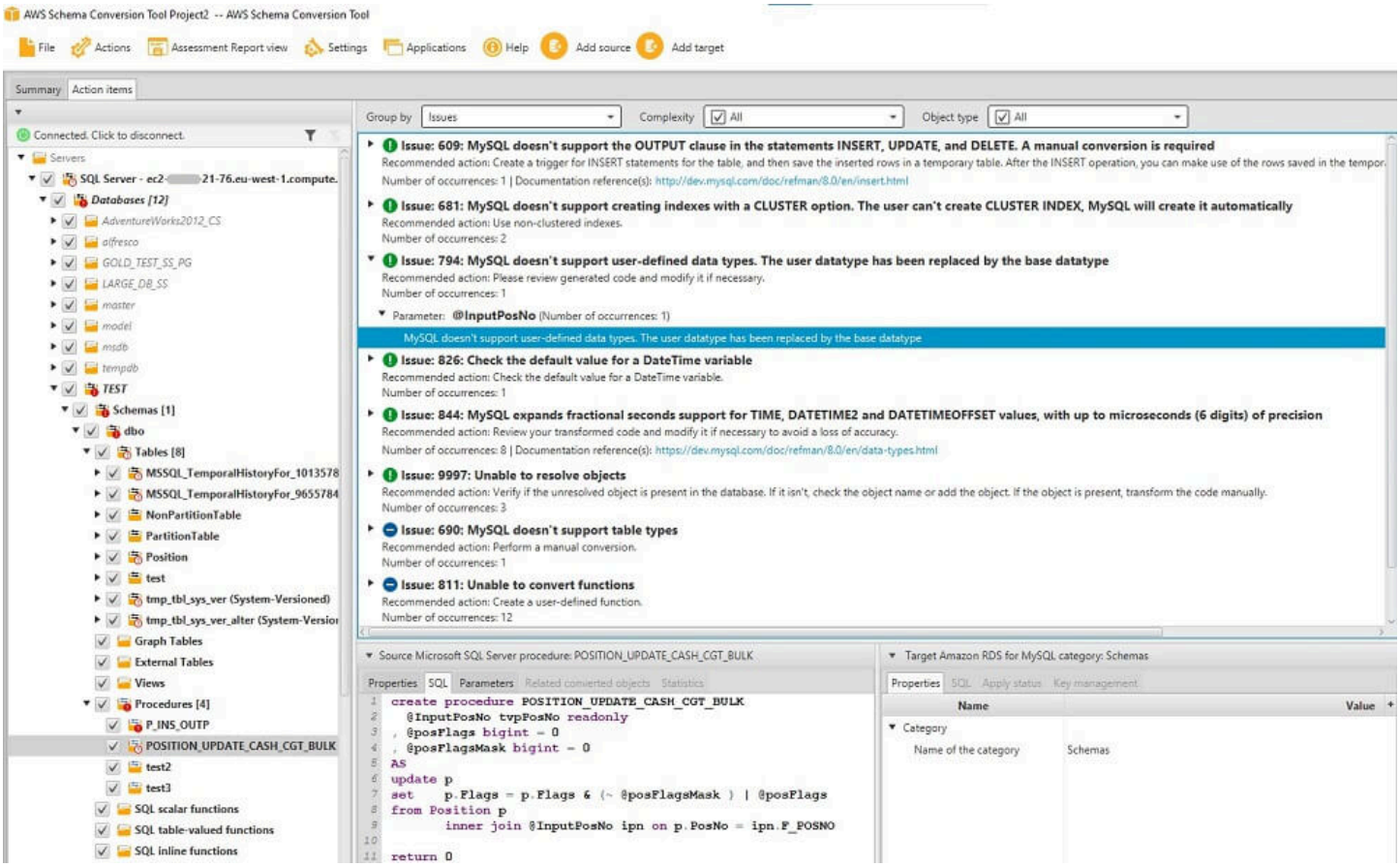
En el informe, el tiempo estimado para convertir estos elementos del esquema se clasifica en los siguientes grupos:

- Simple: acciones que se pueden realizar en menos de una hora.
- Intermedio: acciones que son más complejas y se pueden realizar en el plazo de una a cuatro horas.
- Significativo: acciones que son muy complejas y requieren más de cuatro horas.

Elementos de acción del informe de evaluación

La vista del informe de evaluación incluye también una pestaña Elementos de acción. Esta pestaña contiene una lista de los elementos que no se pueden convertir automáticamente en el motor de base de datos de su base de datos de destino. Si selecciona un elemento de acción de la lista, AWS SCT resalta el elemento en el esquema al que se aplica el elemento de acción.

El informe también contiene recomendaciones sobre cómo convertir el elemento del esquema de forma manual. Para obtener más información sobre decidir cómo administrar las conversiones manuales, consulte Administrar conversiones manuales en AWS SCT.



Guardar el informe de evaluación

Puede guardar una copia local del informe de evaluación de la migración de la base de datos en formato PDF o como un archivo de valores separados por comas (CSV). El archivo CSV contiene solamente información de elementos de acción. El archivo PDF contiene el resumen de acciones e información de los elementos de acción, tal y como se muestra en el siguiente ejemplo.

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 identifed 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.

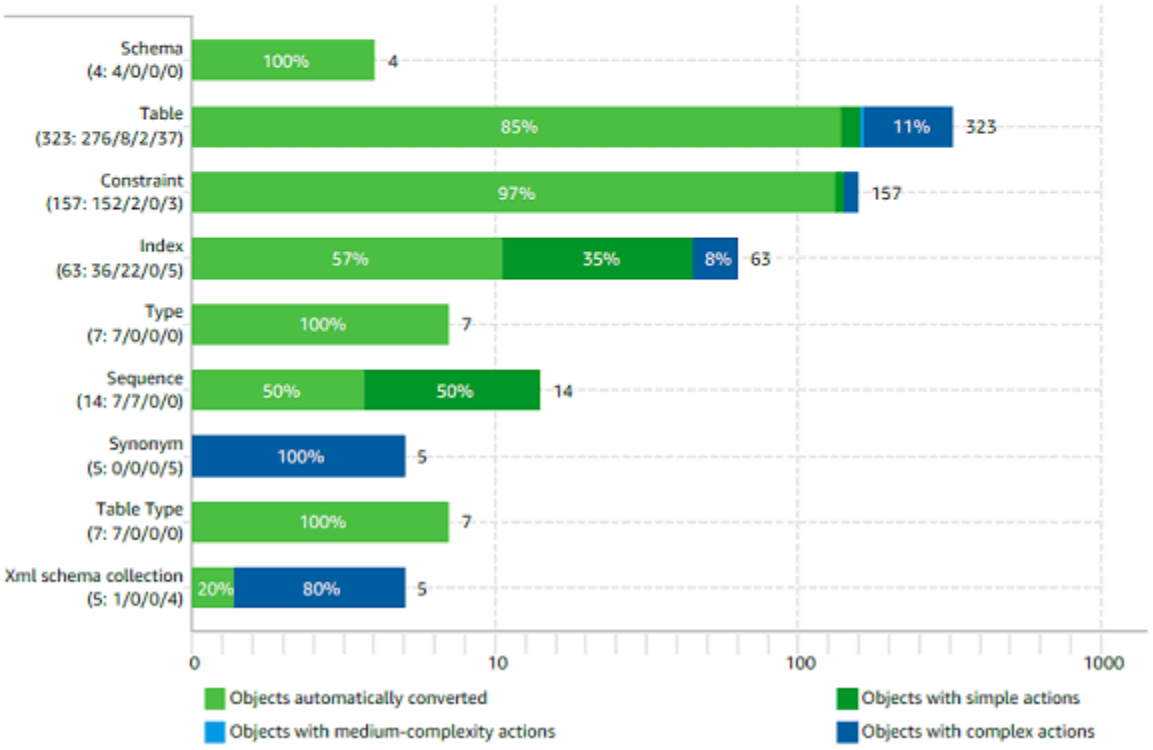
We found 7 encrypted object(s).

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



Administrar conversiones manuales en AWS SCT

El informe de evaluación incluye una lista de los elementos que no se pueden convertir automáticamente en el motor de base de datos de su base de datos de destino. Para cada elemento que no se puede convertir, hay un elemento de acción en la pestaña Elementos de acción.

Puede responder a los elementos de acción del informe de evaluación de las siguientes formas:

- Modificar su esquema de base de datos de origen.

- Modificar su esquema de base de datos de destino.

Modificar su esquema de origen

Para algunos elementos, es posible que resulte más fácil modificar el esquema de base de datos en su esquema de base de datos de origen que lograr que se convierta automáticamente. En primer lugar, verifique que los nuevos cambios son compatibles con la arquitectura de su aplicación y, a continuación, actualice el esquema en la base de datos de origen. Por último, actualice su proyecto con la información de esquema actualizada. Después, puede convertir el esquema actualizado y generar de nuevo el informe de evaluación de la migración de la base de datos. Los elementos de acción ya no aparecen para los elementos que hayan cambiado en el esquema de origen.

La ventaja de este proceso es que el esquema actualizado está siempre disponible al actualizar desde la base de datos de origen.

Modificar su esquema de destino

Para algunos elementos, puede que sea más sencillo aplicar el esquema convertido a la base de datos de destino y, a continuación, agregar manualmente los elementos de esquema equivalentes a la base de datos de destino para los elementos que no se hayan podido convertir automáticamente. Puede escribir todo el esquema que se pueda convertir automáticamente en su base de datos de destino aplicando el esquema. Para obtener más información, consulte Guardar y aplicar un esquema convertido en AWS SCT.

El esquema que escrito en la base de datos de destino no contendrá los elementos que no se hayan podido convertir automáticamente. Después de aplicar el esquema a la base de datos de destino, puede crear manualmente esquemas en dicha base de datos que sean equivalentes a los de la base de datos de origen. Los elementos de acción en el informe de evaluación de la migración de la base de datos contienen sugerencias sobre cómo crear un esquema equivalente.

> **Warning**
>
> Si crea manualmente esquemas en su base de datos de destino, guarde una copia de cualquier operación manual que haya realizado. Si aplica el esquema convertido desde su proyecto a su base de datos de destino de nuevo, se sobrescribirá el trabajo manual que haya realizado.

En algunos casos, no podrá crear un esquema equivalente en la base de datos de destino. Tal vez tenga que rediseñar una parte de la aplicación y de la base de datos para usar la funcionalidad disponible en el motor de base de datos para la base de datos de destino. En otros casos, puede omitir simplemente el esquema que no se pueda convertir automáticamente.

Actualizar un esquema convertido en AWS SCT

Puede actualizar el esquema de origen y el esquema de destino en su proyecto de AWS Schema Conversion Tool.

- Origen: si actualiza el esquema para su base de datos de origen, AWS SCT sustituye el esquema de su proyecto con el último esquema de su base de datos de origen. Con esta funcionalidad, puede actualizar su proyecto si se ha modificado el esquema de la base de datos de origen.
- Destino: si actualiza el esquema para su base de datos de destino, AWS SCT sustituye el esquema de su proyecto con el último esquema de su base de datos de destino. Si no ha aplicado ningún esquema a su base de datos de destino, AWS SCT elimina el esquema convertido de su proyecto. A continuación, puede convertir el esquema de la base de datos de origen a una base de datos de destino limpia.

El esquema en su proyecto de la AWS SCT se actualiza seleccionando Actualizar desde base de datos, como se muestra a continuación.

Guardar y aplicar un esquema convertido en AWS SCT

Cuando la AWS Schema Conversion Tool genera un esquema convertido (tal y como se muestra en Convertir su esquema mediante AWS SCT), no aplica inmediatamente el esquema convertido a la base de datos de destino. Los esquemas convertidos se almacenan en el entorno local de su proyecto hasta que esté listo para aplicarlos en la base de datos de destino. Con esta funcionalidad, puede trabajar con los elementos de esquema que no se puedan convertir automáticamente en su motor de base de datos de destino. Para obtener más información sobre elementos que no se pueden convertir automáticamente, consulte Crear informes de evaluación de migración con AWS SCT.

Si lo desea, puede hacer que la herramienta guarde su esquema convertido en un archivo como script SQL antes de aplicar el esquema a su base de datos de destino. También puede hacer que la herramienta aplique el esquema convertido directamente a su base de datos de destino.

Guardar el esquema convertido en un archivo

Puede guardar los esquemas convertidos como scripts SQL en un archivo de texto. Al utilizar este enfoque, puede modificar los scripts SQL generados en la AWS SCT para abordar los elementos que no puedan convertirse automáticamente. A continuación, puede ejecutar sus scripts actualizados en su instancia de base de datos de destino para aplicar el esquema convertido a la base de datos de destino.

Para guardar el esquema convertido como scripts de SQL

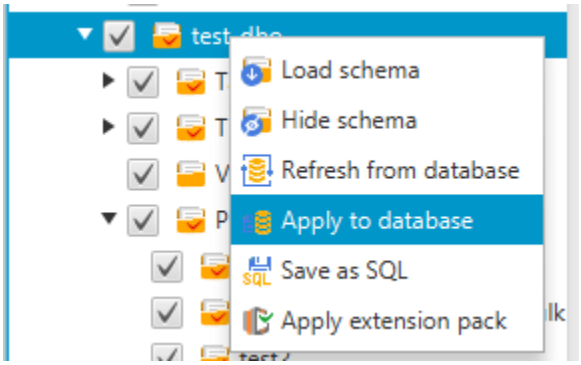
1. Elija su esquema y abra el menú contextual (clic secundario).
2. Elija Guardar como.
3. Introduzca el nombre del archivo y seleccione Guardar.
4. Guarde el esquema convertido si opta por una de las opciones siguientes:
   - Archivo único
   - Archivo único por fase
   - Archivo único por instrucción

Para elegir el formato del script de SQL

1. En el menú Configuración, seleccione Configuración del proyecto.
2. Seleccione Guardar scripts.
3. En Proveedor, elija la plataforma de base de datos.
4. En Guardar scripts de SQL en, elija cómo desea guardar el script de esquema de base de datos.
5. Seleccione Aceptar para guardar la configuración.

Aplicar el esquema convertido

Cuando esté listo para aplicar el esquema convertido a la base de datos de destino, elija el elemento del esquema en el panel derecho del proyecto. Abra el menú contextual (clic secundario) del elemento del esquema y seleccione Aplicar a base de datos, como se muestra a continuación.



El esquema del paquete de extensión

La primera vez que aplique el esquema convertido a su instancia de base de datos de destino, la AWS SCT agrega un esquema adicional a su instancia de base de datos de destino. Este esquema implementa las funciones del esquema de la base de datos que son necesarias a la hora de escribir el esquema convertido en la instancia de base de datos de destino. El esquema se denomina esquema de paquete de extensión.

No modifique el esquema del paquete de extensión, ya que podría encontrarse resultados imprevistos en el esquema convertido creado en la instancia de base de datos de destino. Cuando el esquema haya migrado totalmente a su instancia de base de datos de destino y ya no necesite la AWS SCT, puede eliminar el esquema del paquete de extensión.

El esquema de paquete de extensión se denomina en función de su base de datos de origen, de la siguiente manera:

- Greenplum: `aws_greenplum_ext`
- Microsoft SQL Server: `aws_sqlserver_ext`
- Netezza: `aws_netezza_ext`
- Oracle: `aws_oracle_ext`
- Snowflake: `aws_snowflake_ext`
- Teradata: `aws_teradata_ext`
- Vertica: `aws_vertica_ext`

Para obtener más información, consulte Uso de paquetes AWS SCT de extensión.

Bibliotecas Python

Para crear funciones personalizadas en Amazon Redshift, utilice el lenguaje Python. Utilice el paquete de extensión de AWS SCT para instalar bibliotecas python para su base de datos de Amazon Redshift. Para obtener más información, consulte Uso de paquetes AWS SCT de extensión.

Optimizar Amazon Redshift mediante AWS SCT

Puede utilizar AWS Schema Conversion Tool para optimizar su base de datos de Amazon Redshift. AWS SCT recomienda claves de clasificación y distribución para optimizar su base de datos, utilizando su base de datos de Amazon Redshift como origen y una base de datos de Amazon Redshift de prueba como destino.

Optimizar la base de datos de Amazon Redshift

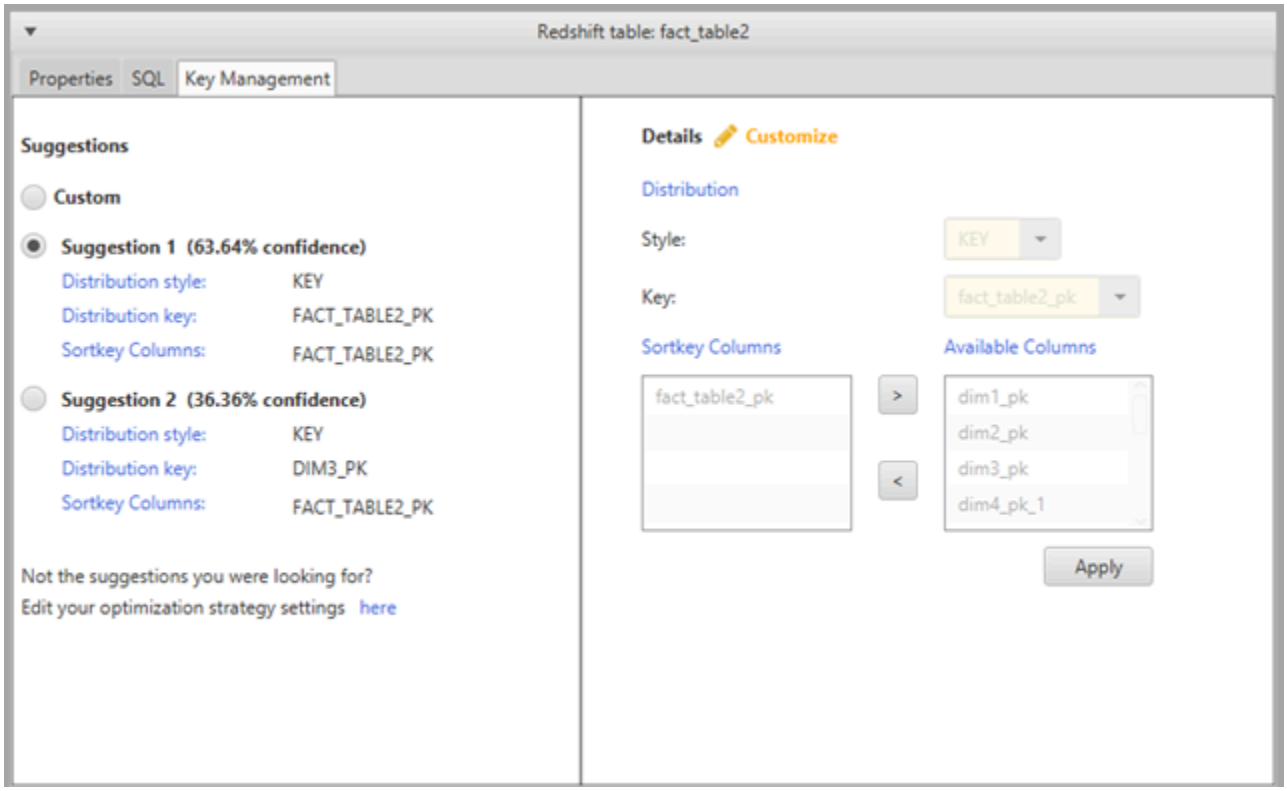
Utilice el siguiente procedimiento para optimizar su base de datos de Amazon Redshift.

Para optimizar la base de datos de Amazon Redshift

1. Tome una instantánea manual de su clúster de Amazon Redshift como copia de seguridad. Puede eliminar la instantánea después de que haya terminado la optimización de su clúster de Amazon Redshift y probar los cambios efectuados. Para obtener más información, consulte Amazon Redshift Snapshots.
2. Seleccione un objeto de esquema para convertir en el panel izquierdo del proyecto. Abra el menú contextual (clic secundario) para el objeto y seleccione Recopilar estadísticas.

   AWS SCT utiliza las estadísticas para realizar sugerencias para las claves de ordenación y distribución.
3. Seleccione un objeto de esquema para optimizar en el panel izquierdo del proyecto. Abra el menú contextual (clic secundario) para el objeto y seleccione Ejecutar optimización.

   AWS SCT realiza sugerencias para las claves de ordenación y distribución.
4. Para revisar las sugerencias, expanda el nodo de tablas de su esquema en el panel izquierdo de su proyecto y, a continuación, elija una tabla. Seleccione la pestaña Administración de claves como se muestra a continuación.



El panel izquierdo contiene sugerencias claves, e incluye la puntuación de fiabilidad para cada sugerencia. Puede seleccionar una de las sugerencias, o personalizar la clave editándola en el panel derecho.

5. Puede crear un informe que contenga las sugerencias de optimización. Para crear el informe, haga lo siguiente:

   a. Seleccione un objeto de esquema que haya optimizado en el panel izquierdo del proyecto. Abra el menú contextual (clic secundario) para el objeto y seleccione Crear informe.

      El informe se abrirá en la ventana principal y aparecerá la pestaña Resumen. El número de objetos con sugerencias de optimización aparecerá en el informe.

   b. Seleccione la pestaña Action Items para ver las principales sugerencias en un formato de informe.

   c. Puede guardar una copia local del informe de optimización en formato PDF o como un archivo de valores separados por comas (CSV). El archivo CSV contiene solamente información de elementos de acción. El archivo PDF contiene tanto el resumen como información de elementos de acción.

6. Para aplicar las optimizaciones sugeridas en la base de datos, seleccione un objeto en el panel derecho de su proyecto. Abra el menú contextual (clic con el botón secundario) del objeto y seleccione Aplicar a base de datos.

Conversión de procesos de extracción, transformación y carga (ETL) con AWS Schema Conversion Tool

Puede utilizar AWS Schema Conversion Tool (AWS SCT) para migrar procesos de extracción, transformación y carga (ETL). Este tipo de migración incluye la conversión de la lógica empresarial relacionada con ETL. Esta lógica puede residir en el almacenamiento de datos de origen o en scripts externos que se ejecutan por separado.

Actualmente, AWS SCT admite la conversión de scripts de ETL en objetos para RSQL de AWS Glue y Amazon Redshift, como se muestra en la siguiente tabla.

Origen	Destino
Scripts de ETL de Informatica	Informatica
Paquetes de ETL de Microsoft SQL Server Integration Services (SSIS)	AWS Glue o AWS Glue Studio
Scripts de intérprete de comandos con comandos incrustados de Teradata Basic Teradata Query (BTEQ)	RSQL de Amazon Redshift
Scripts de ETL de Teradata BTEQ	RSQL de AWS Glue o Amazon Redshift
Scripts de trabajo de Teradata FastExport	RSQL de Amazon Redshift
Scripts de trabajo de Teradata FastLoad	RSQL de Amazon Redshift
Scripts de trabajo de Teradata MultiLoad	RSQL de Amazon Redshift

Temas

- Conversión de procesos de ETL a AWS Glue con AWS SCT
- Conversión de procesos de ETL mediante la API de Python para AWS Glue con AWS SCT
- Convertir los scripts de ETL de Informatica con AWS SCT
- Conversión de SSIS a AWS Glue con AWS SCT

- Convertir SSIS a AWS Glue Studio con AWS SCT
- Convertir scripts de BTEQ de Teradata a RSQL de Amazon Redshift AWS SCT
- Conversión de scripts del intérprete de comandos con comandos BTEQ de Teradata incrustados a RSQL de Amazon Redshift con AWS SCT
- Convertir scripts de trabajo de FastExport de Teradata a RSQL de Amazon Redshift con AWS SCT
- Conversión de scripts de trabajo de FastLoad de Teradata a RSQL de Amazon Redshift con AWS SCT
- Convertir scripts de trabajo de MultiLoad de Teradata a RSQL de Amazon Redshift con AWS SCT

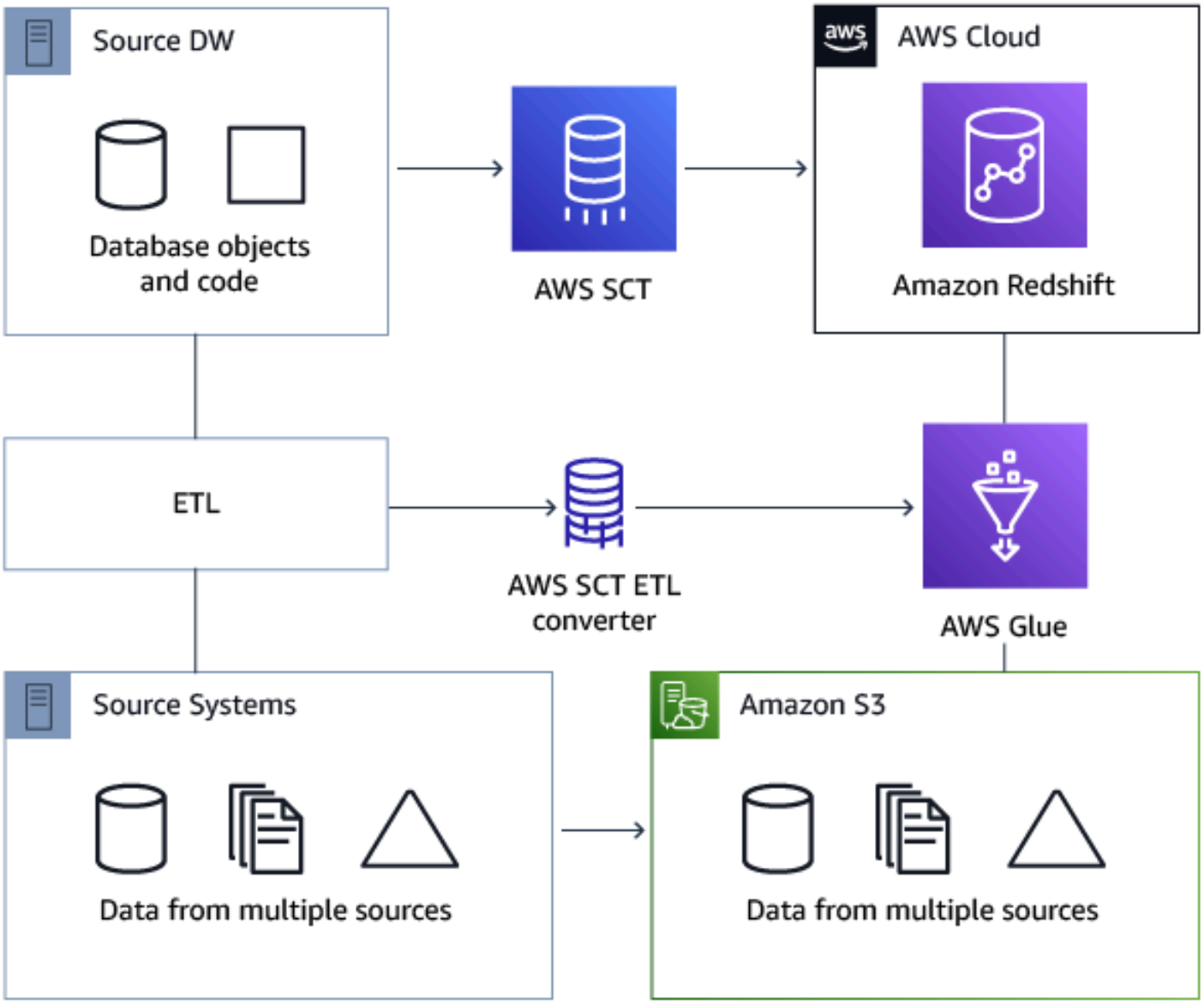
Conversión de procesos de ETL a AWS Glue con AWS SCT

A continuación encontrará una descripción del proceso a seguir para convertir scripts de ETL a AWS Glue con AWS SCT. En este ejemplo vamos a convertir una base de datos Oracle a Amazon Redshift, junto con los procesos de ETL utilizados con las bases de datos de origen y almacenamientos de datos.

Temas

- Requisitos previos
- Descripción de AWS Glue Data Catalog
- Limitaciones para la conversión mediamte AWS SCT con AWS Glue
- Paso 1: Crear un nuevo proyecto de
- Paso 2: Crear un trabajo de AWS Glue

El siguiente diagrama de arquitectura muestra un ejemplo de proyecto de migración de bases de datos que incluye la conversión de scripts de ETL a AWS Glue.



Requisitos previos

Antes de comenzar, haga lo siguiente:

- Migre las bases de datos de origen que desee migrar a AWS.
- Migre los almacenamientos de datos de destino a AWS.
- Recopile una lista de todo el código involucrado en su proceso de ETL.
- Recopile una lista de toda la información de conexión necesaria para cada base de datos.

Además, AWS Glue necesita permisos para obtener acceso a otros recursos de AWS en su nombre. Estos permisos los concede utilizando AWS Identity and Access Management (IAM). Asegúrese de haber creado una política de IAM para AWS Glue. Para obtener más información, consulte Crear una política de IAM para el servicio de AWS Glue en la Guía para desarrolladores de AWS Glue.

Descripción de AWS Glue Data Catalog

Como parte del proceso de conversión, AWS Glue carga la información sobre las bases de datos de origen y destino. Además, organiza esta información en categorías, en una estructura que se denomina árbol. La estructura incluye lo siguiente:

- Conexiones: parámetros de conexión
- Rastreadores: una lista de rastreadores, un rastreador para cada esquema
- Bases de datos: contenedores que contienen tablas
- Tablas: definiciones de metadatos que representan los datos de las tablas
- Trabajos de ETL: lógica empresarial que realiza el trabajo de ETL
- Activadores: lógica que controla cuándo se ejecuta un trabajo de ETL en AWS Glue (ya sea a petición, según lo programado o activado por eventos del trabajo)

AWS Glue Data Catalog es un índice para las métricas de tiempo de ejecución, esquema y ubicación de sus datos. Al trabajar con AWS Glue y AWS SCT, AWS Glue Data Catalog incluye referencias a datos que se utilizan como orígenes y destinos de sus trabajos de ETL en AWS Glue. Para crear su almacenamiento de datos, catalogue estos datos.

Puede usar la información del Catálogo de datos para crear y monitorizar sus trabajos de ETL. Normalmente, deberá ejecutar un rastreador para realizar un inventario de los datos incluidos en sus almacenes de datos, pero existen otras formas de añadir tablas de metadatos en el Catálogo de datos.

Al definir una tabla en su Data Catalog, puede añadirla a una base de datos. Una base de datos se usa para organizar tablas en AWS Glue.

Limitaciones para la conversión mediamte AWS SCT con AWS Glue

Al convertir mediante AWS SCT con AWS Glue se aplican las siguientes limitaciones:

Recurso	Límite predeterminado

Número de bases de datos para cada cuenta	10 000
Número de tablas para cada base de datos	100 000
Número de particiones para cada tabla	1 000 000
Número de versiones de tabla para cada tabla	100 000
Número de tablas para cada cuenta	1 000 000
Número de particiones para cada cuenta	10 000 000
Número de versiones de tabla para cada cuenta	1 000 000
Número de conexiones para cada cuenta	1 000
Número de rastreadores para cada cuenta	25
Número de trabajos para cada cuenta	25
Número de disparadores para cada cuenta	25
Número de ejecuciones de trabajo simultáneas para cada cuenta	30
Número de ejecuciones de trabajo simultáneas para cada trabajo	3
Número de trabajos para cada disparador	10
Número de puntos de enlace de desarrollo para cada cuenta	5
Máximo de unidades de procesamiento de datos (DPU) utilizados por un punto de conexión de desarrollo a la vez	5
DPU máximos utilizados por un rol a la vez	100

Longitud del nombre de la base de datos	Sin límite Para que sea compatible con otros almacenes de metadatos, como Apache Hive, el nombre se incorpora en minúsculas. Si tiene previsto obtener acceso a la base de datos desde Amazon Athena, proporcione un nombre únicamente con caracteres alfanumér icos y guiones bajos.
Longitud del nombre de la conexión	Sin límite
Longitud del nombre del rastreador	Sin límite

Paso 1: Crear un nuevo proyecto de

Para crear un proyecto nuevo, siga estos pasos generales:

1. Cree un proyecto nuevo en AWS SCT. Para obtener más información, consulte Creación de un proyecto de AWS SCT.
2. Agregue sus bases de datos de origen y destino al proyecto. Para obtener más información, consulte Cómo agregar servidores de bases de datos a un proyecto de AWS SCT.

   Asegúrese de haber elegido Usar AWS Glue en la configuración de conexión a la base de datos de destino. Para ello, seleccione la pestaña AWS Glue. En Copiar desde el perfil de AWS, elija el perfil que desea utilizar. El perfil debe rellenar automáticamente la clave de acceso de AWS, la clave secreta y la carpeta de bucket de Amazon S3. Si no lo hace, indique dicha información. Después de seleccionar Aceptar, AWS Glue analiza los objetos y carga los metadatos en el AWS Glue Data Catalog.

   Según la configuración de seguridad, es posible que aparezca un mensaje de advertencia que indique que su cuenta no tiene privilegios suficientes para algunos de los esquemas del servidor. Si tiene acceso a los esquemas que está utilizando, puede ignorar este mensaje.
3. Para finalizar la preparación de la importación de su ETL, establezca conexiones con sus bases de datos de origen y de destino. Para ello, elija la base de datos en el árbol de metadatos de origen o destino y, a continuación, seleccione Conectar al servidor.

AWS Glue crea una base de datos en el servidor de base de datos de origen y otra en el servidor de base de datos de destino para ayudar en la conversión de ETL. La base de datos en el servidor de destino contiene el AWS Glue Data Catalog. Para encontrar objetos específicos, utilice el botón de búsqueda en los paneles de origen o destino.

Para ver cómo se convierte un objeto específico, busque un elemento que desee convertir y seleccione Convertir esquema desde el menú contextual (clic secundario). AWS SCT transforma este objeto seleccionado en un script.

Puede revisar el script convertido desde la carpeta Scripts del panel derecho. Actualmente, el script es un objeto virtual, que solo está disponible como parte de su proyecto de AWS SCT.

Para crear un trabajo de AWS Glue con el script convertido, cárguelo en Amazon S3. Para cargar el script a Amazon S3, selecciónelo y, a continuación, elija Guardar en S3 en el menú contextual (clic secundario).

Paso 2: Crear un trabajo de AWS Glue

Después de guardar el script en Amazon S3, puede seleccionarlo y elegir Configurar trabajo de AWS Glue para abrir el asistente para configurar el trabajo de AWS Glue. El asistente facilita esta configuración:

1. La primera pestaña del asistente, Diseñar flujo de datos, le permite elegir una estrategia de ejecución y la lista de scripts que desea incluir en este trabajo. Puede elegir los parámetros de cada script. También puede reorganizar los scripts de manera que se ejecuten en el orden correcto.
2. En la segunda pestaña, puede asignar un nombre al trabajo y configurar directamente las opciones para AWS Glue. En esta pantalla puede configurar las siguientes opciones:
   - Rol de AWS Identity and Access Management (IAM)
   - Nombres de archivos de script y rutas de archivo
   - Cifre el script usando cifrado del lado del servidor con claves administradas por Amazon S3 (SSE-S3)
   - Directorio temporal
   - Ruta de la biblioteca Python generada
   - Ruta de la biblioteca Python del usuario
   - Ruta de los archivos .jar dependientes
   - Ruta de archivos a la que se hace referencia

- DPU simultáneas para cada trabajo ejecutado
- Simultaneidad máxima
- Tiempo de espera del trabajo (en minutos)
- Umbral de notificación de retraso (en minutos)
- Number of retries (Número de reintentos)
- Configuración de seguridad
- Cifrado en el servidor

3. En el tercer paso, o pestaña, elija la conexión configurada con el punto de enlace de destino.

Tras finalizar la configuración del trabajo, este se muestra en los trabajos de ETL en el AWS Glue Data Catalog. Si elige el trabajo, se mostrará la configuración para que pueda revisarla o editarla. Para crear un nuevo trabajo en AWS Glue, elija Crear trabajo de AWS Glue en el menú contextual (clic secundario) del trabajo. Al hacerlo se aplica la definición de esquema. Para actualizar la visualización, elija Actualizar desde la base de datos en el menú contextual (clic secundario).

En este momento, podrá ver su trabajo en la consola de AWS Glue. Para ello, inicie sesión en AWS Management Console y abra la consola de AWS Glue en https://console.aws.amazon.com/glue/.

Puede probar el nuevo trabajo para asegurarse de que funciona correctamente. Compruebe los datos de la tabla de origen y, a continuación, verifique que la tabla de destino esté vacía. Ejecute el trabajo y vuelva a realizar la comprobación. Puede ver los registros de errores desde la consola de AWS Glue.

Conversión de procesos de ETL mediante la API de Python para AWS Glue con AWS SCT

En las secciones siguientes se describe una conversión que llama a las operaciones de la API de AWS Glue en Python. Para obtener más información, consulte Programar scripts de AWS Glue en Python en la Guía para desarrolladores de AWS Glue.

Temas

- Paso 1: Crear una base de datos
- Paso 2: Crear una conexión
- Paso 3: Crear un rastreador de AWS Glue

Paso 1: Crear una base de datos

El primer paso consiste en crear una base de datos nueva en un AWS Glue Data Catalog utilizando la AWSAPI del SDK de . Al definir una tabla en el Data Catalog, la añade a una base de datos. Las bases de datos se usan para organizar tablas en AWS Glue.

En el ejemplo siguiente se muestra el método `create_database` de la API de Python para AWS Glue.

```
response = client.create_database(
    DatabaseInput={
        'Name': 'database_name',
        'Description': 'description',
        'LocationUri': 'string',
        'Parameters': {
            'parameter-name': 'parameter value'
        }
    }
)
```

Si utiliza Amazon Redshift, el nombre de la base de datos se forma tal y como se indica a continuación.

```
{redshift_cluster_name}_{redshift_database_name}_{redshift_schema_name}
```

El nombre completo del clúster de Amazon Redshift de este ejemplo es el siguiente.

```
rsdbb03.apq1mpqso.us-west-2.redshift.amazonaws.com
```

A continuación, se muestra un ejemplo de un nombre correcto de la base de datos. En este caso `rsdbb03` es el nombre, que es la primera parte del nombre completo del punto de enlace del clúster. La base de datos se denomina `dev` y el esquema es `ora_glue`.

```
rsdbb03_dev_ora_glue
```

Paso 2: Crear una conexión

Cree una nueva conexión en un Data Catalog utilizando la API del SDK de AWS.

En el ejemplo siguiente se muestra el uso del método `create_connection` de la API de Python para AWS Glue.

```
response = client.create_connection(
    ConnectionInput={
        'Name': 'Redshift_abcde03.aabbcc112233.us-west-2.redshift.amazonaws.com_dev',
        'Description': 'Created from SCT',
        'ConnectionType': 'JDBC',
        'ConnectionProperties': {
            'JDBC_CONNECTION_URL': 'jdbc:redshift://aabbcc03.aabbcc112233.us-
west-2.redshift.amazonaws.com:5439/dev',
            'USERNAME': 'user_name',
            'PASSWORD': 'password'
        },
        'PhysicalConnectionRequirements': {
            'AvailabilityZone': 'us-west-2c',
            'SubnetId': 'subnet-a1b23c45',
            'SecurityGroupIdList': [
                'sg-000a2b3c', 'sg-1a230b4c', 'sg-aba12c3d', 'sg-1abb2345'
            ]
        }
    }
)
```

Los parámetros que se usan en `create_connection` son los siguientes:

- `Name` (cadena UTF-8): obligatorio. En Amazon Redshift, el nombre de la conexión se forma tal y como se indica a continuación: `Redshift_<Endpoint-name>_<redshift-database-name>`, por ejemplo: `Redshift_abcde03_dev`
- `Description` (cadena UTF-8): su descripción de la conexión.
- `ConnectionType` (cadena UTF-8): obligatorio. El tipo de conexión. En la actualidad, solo se admite JDBC; SFTP es incompatible.
- `ConnectionProperties` (dict): obligatorio. Una lista de pares clave-valor utilizados como parámetros para esta conexión, incluida la URL de la conexión JDBC, el nombre de usuario y la contraseña.
- `PhysicalConnectionRequirements` (dict): requisitos de conexión física, que incluyen lo siguiente:
  - `SubnetId` (cadena UTF-8): el ID de la subred utilizada por la conexión.

- `SecurityGroupIdList` (lista): el ID del grupo de seguridad usado por la conexión.
- `AvailabilityZone` (cadena UTF-8): obligatorio. La zona de disponibilidad que contiene el punto de conexión. Este parámetro se ha quedado obsoleto.

Paso 3: Crear un rastreador de AWS Glue

A continuación, cree un rastreador de AWS Glue para rellenar el catálogo de AWS Glue. Para obtener más información, consulte la sección de catalogación de tablas con un rastreador en la Guía para desarrolladores de AWS Glue.

El primer paso para añadir un rastreador consiste en crear una nueva base de datos en un Data Catalog utilizando la API del SDK de AWS. Antes de comenzar, primero debe eliminar cualquier versión anterior de la misma versión con la operación `delete_crawler`.

Al crear el rastreador, hay que tener algunas consideraciones en cuenta:

- Para el nombre del rastreador, utilice el formato *<redshift_node_name>*_*<redshift_database_name>*_*<redshift_shema_name>*, por ejemplo: `abcde03_dev_ora_glue`.
- Utilice un rol de IAM que ya existe. Para obtener más información sobre la creación de roles, consulte Crear roles de IAM en la Guía del usuario de IAM.
- Utilice el nombre de la base de datos que creó en los pasos anteriores.
- Utilice el parámetro `ConnectionName`, que es obligatorio.
- Para el parámetro `path`, utilice la ruta al destino de JDBC, por ejemplo: `dev/ora_glue/%`.

En el siguiente ejemplo se elimina un rastreador existente y, a continuación, se crea uno nuevo con la API de Python para 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'
)
```

Después de crear el rastreador, ejecute un rastreador que se conecte a uno o varios almacenes de datos, determine las estructuras de datos y escriba tablas en el Data Catalog. Puede ejecutar su rastreador de manera programada, tal y como se muestra a continuación.

```
response = client.start_crawler(
    Name='string'
)
```

En este ejemplo, se utiliza Amazon Redshift como objetivo. Después de que se ejecute el rastreador, los tipos de datos de Amazon Redshift se asignan al tipo de datos de AWS Glue de la siguiente manera.

Tipos de datos de Amazon Redshift	Tipo de datos de AWS Glue
smallint	smallint
integer	int
bigint	bigint
decimal	decimal(18,0)
decimal(p,s)	decimal(p,s)
real	double
double precision	double
booleano	booleano
char	string
varchar	string
varchar(n)	string
date	date
timestamp	timestamp
timestamptz	timestamp

Convertir los scripts de ETL de Informatica con AWS SCT

Puede utilizar la interfaz de la línea de comandos (CLI) de AWS SCT para convertir los scripts de ETL de Informatica y usarlos con la nueva base de datos de destino. Esta conversión incluye tres pasos clave. En primer lugar, AWS SCT convierte el código SQL que está incrustado en los objetos de Informatica. A continuación, AWS SCT cambia los nombres de los objetos de la base de datos según las reglas de migración que especificó en el proyecto. Por último, AWS SCT redirige las conexiones de los scripts de ETL de Informatica a la nueva base de datos de destino.

Puede convertir los scripts de ETL de Informatica como parte de un proyecto de conversión de bases de datos de AWS SCT. Agregue las bases de datos de origen y destino al proyecto al convertir los scripts de ETL de Informatica.

Para convertir los scripts de ETL de Informatica, utilice la versión 1.0.667 de AWS SCT o versiones posteriores. Además, familiarícese con la interfaz de la línea de comandos de AWS SCT. Para obtener más información, consulte AWS SCT Referencia CLI.

Para convertir los scripts de ETL de Informatica mediante AWS SCT

1. Cree un script de la CLI de AWS SCT nuevo o edite una plantilla de escenario existente. Por ejemplo, puede descargar y editar la plantilla de `InformaticConversionTemplate.scts`. Para obtener más información, consulte Obtención de escenarios de la CLI.
2. Descargue los controladores JDBC necesarios para sus bases de datos de origen y destino. Especifique la ubicación de estos controladores mediante el comando `SetGlobalSettings`. Además, especifique las carpetas en las que AWS SCT puede guardar los archivos de registro.

   El siguiente ejemplo de código muestra cómo agregar la ruta de los controladores Oracle y PostgreSQL a la configuración de AWS SCT. Tras ejecutar este ejemplo de código, AWS SCT almacena los archivos de registro en la carpeta `C:\sct_log`. Además, AWS SCT almacena los archivos de registro de la consola en la carpeta `C:\Temp\oracle_postgresql`.

   ```
   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. Cree un proyecto de AWS SCT nuevo. Introduzca el nombre y la ubicación del proyecto.

   El siguiente ejemplo de código crea el proyecto de `oracle_postgresql` en la carpeta `C:\Temp`.

```
CreateProject
 -name: 'oracle_postgresql'
 -directory: 'C:\Temp'
/
```

4. Agregue información de conexión sobre las bases de datos de origen y destino.

   El siguiente ejemplo de código agrega bases de datos Oracle y PostgreSQL como origen y destino del proyecto de 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'
/
```

   En el ejemplo anterior, sustituya *source_user* y *target_user* por los nombres de los usuarios de las bases de datos. A continuación, sustituya *source_password* y *target_password* por las contraseñas correspondientes. Para *source_address* y *target_address*, introduzca las direcciones IP de los servidores de las bases de datos de origen y destino.

   Para conectarse a una base de datos Oracle de la versión 19 o posterior, utilice el nombre del servicio de Oracle en el comando `AddSource`. Para ello, agregue el parámetro `-connectionType` y establezca su valor en `'basic_service_name'`. A continuación, añada el parámetro `-servicename` y establezca su valor en el nombre de su servicio de Oracle. Para

obtener más información sobre el comando `AddSource`, consulte la Referencia de la CLI de AWS Schema Conversion Tool.

5. Cree una regla de asignación de AWS SCT nueva, que defina los motores de la base de datos de destino para cada esquema de base de datos de origen. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT.

   El siguiente ejemplo de código crea una regla de asignación que incluye todos los esquemas de bases de datos Oracle de origen y define PostgreSQL como destino de migración.

   ```
   AddServerMapping
    -sourceTreePath: 'Servers.ORACLE'
    -targetTreePath: 'Servers.POSTGRESQL'
   /
   ```

6. Agregue información de conexión sobre los archivos XML de origen y destino de Informatica.

   En el siguiente ejemplo de código se añaden los archivos XML de Informatica de las carpetas `C:\Informatica_source` y `C:\Informatica_target`.

   ```
   AddSource
    -name: 'INFA_SOURCE'
    -vendor: 'INFORMATICA'
    -mappingsFolder: 'C:\Informatica_source'
   /
   AddTarget
    -name: 'INFA_TARGET'
    -vendor: 'INFORMATICA'
    -mappingsFolder: 'C:\Informatica_target'
   /
   ```

7. Cree otra regla de asignación para definir el archivo XML de destino de Informatica para el archivo XML de origen de Informatica.

   El siguiente ejemplo de código crea una regla de asignación que incluye los archivos XML de origen y destino de Informatica utilizados en el ejemplo anterior.

   ```
   AddServerMapping
   -sourceTreePath: 'ETL.INFA_SOURCE'
   -targetTreePath: 'ETL.INFA_TARGET'
   /
   ```

8. Especifique la conexión al servidor de base de datos que corresponde a la referencia del nombre de conexión de Informatica.

   El siguiente ejemplo de código configura la redirección de los scripts de ETL de Informatica desde la base de datos de origen a la nueva base de datos de destino. En este ejemplo también se configuran las variables de conexión.

```
ConfigureInformaticaConnectionsRedirect
 -treePath: 'ETL.INFA_SOURCE.Files'
 -connections: '{
 "ConnectionNames": [
 {
  "name": "Oracle_src",
  "newName": "postgres",
  "treePath": "Servers.ORACLE"
 }
 ]
 "ConnectionVariables": [
 {
         "name": "$Source",
         "treePath": "Servers.ORACLE"
    }
    ]
 }'
/
```

9. Convierta los esquemas de la base de datos de origen y los scripts de ETL de Informatica.

   En el siguiente ejemplo de código se convierten todos los esquemas de bases de datos Oracle de origen y el archivo XML de Informatica.

```
Convert
 -treePath: 'Servers.ORACLE.Schemas.%'
/
Convert
 -treePath: 'ETL.INFA_SOURCE.Files'
/
```

10. (Opcional) Guarde el proyecto de conversión y el informe de evaluación. En este informe se incluyen los elementos de acción de la conversión y recomendaciones sobre cómo abordarlas.

El siguiente ejemplo de código guarda su proyecto y una copia del informe de evaluación en un archivo PDF en la carpeta `C:\Temp`.

```
SaveProject
/
SaveReportPDF
 -treePath: 'ETL.INFA_SOURCE.Files'
 -file:'C:\Temp\Informatica.pdf'
/
```

11. Guarde el archivo XML convertido de Informatica.

    En el siguiente ejemplo de código se guarda el archivo XML convertido en la carpeta `C:\Temp`. Usted especificó esta carpeta en el paso anterior mediante el comando `AddTarget`.

    ```
    SaveTargetInformaticaXML
    -treePath: 'ETL.INFA_TARGET.Files'
    /
    ```

12. Guarde el script como un archivo de `.scts` y ejecútelo mediante el comando `RunSCTBatch` de la CLI de AWS SCT. Para obtener más información, consulte AWS SCT Modo de script CLI.

    En el ejemplo siguiente se ejecuta el script de `Informatica.scts` en la carpeta `C:\Temp`. Puede usar este ejemplo en Windows.

    ```
    RunSCTBatch.cmd --pathtoscts "C:\Temp\Informatica.scts"
    ```

    Si edita los scripts de ETL de Informatica de origen, vuelva a ejecutar el script de la CLI de AWS SCT.

Conversión de SSIS a AWS Glue con AWS SCT

A continuación se explica cómo convertir paquetes de Microsoft SQL Server Integration Services (SSIS) a AWS Glue mediante AWS SCT.

Para convertir paquetes de Microsoft SSIS a AWS Glue, utilice la versión 1.0.642 de AWS SCT o una versión posterior. También debe tener un proyecto de SSIS con paquetes de ETL (archivos `.dtsx`, `.conmgr` y `.params`) en la carpeta local.

No es necesario instalar un servidor SSIS. El proceso de conversión recorre los archivos de SSIS locales.

Para convertir un paquete de SSIS a AWS Glue mediante AWS SCT

1. Cree un proyecto nuevo en AWS SCT o abra un proyecto existente. Para obtener más información, consulte the section called “Creación de un proyecto”.
2. Seleccione Agregar origen en el menú para añadir un paquete de SSIS de origen nuevo al proyecto.
3. Elija SQL Server Integration Services y complete lo siguiente:

   - Nombre de conexión: introduzca el nombre de la conexión. AWS SCT muestra este nombre en el árbol de metadatos.
   - Carpeta de paquetes de SSIS: elija la ruta a la carpeta del proyecto de SSIS en la que estén los paquetes.

   AWS SCT lee los archivos del proyecto (archivos con las extensiones `.dtsx`, `.conmgr` o `.params`) de la carpeta local y los analiza. A continuación, los organiza en un árbol de categorías de AWS SCT.
4. Seleccione Agregar destino en el menú para añadir una plataforma de destino nueva para convertir los paquetes de SSIS de origen.
5. Elija AWS Glue y complete lo siguiente:

   - Nombre de conexión: introduzca el nombre de la conexión. AWS SCT muestra este nombre en el árbol de metadatos.
   - Copiar del perfil de AWS: elija el perfil que desee utilizar.
   - Clave de acceso de AWS: introduzca la clave de acceso de AWS.
   - Clave secreta de AWS: introduzca su clave secreta de AWS.
   - Región: seleccione la Región de AWS que quiera utilizar de la lista.
   - Carpeta del bucket de Amazon S3: introduzca la ruta de la carpeta del bucket de Amazon S3 que va a utilizar.

   Puede utilizar un destino de AWS Glue virtual. En este caso, no es necesario especificar las credenciales de conexión. Para obtener más información, consulte the section called “Destinos virtuales”.

6. Cree una nueva regla de asignación que incluya su paquete de SSIS de origen y su destino de AWS Glue. Para obtener más información, consulte the section called “Regla nueva”.
7. En el menú Ver, seleccione Vista principal.
8. En la vista de árbol de SSIS, abra el menú contextual (clic secundario) de Administradores de conexiones y, a continuación, elija Configurar conexiones.
9. Configure el administrador de conexiones del proyecto.

   Para configurar una asignación de conexiones para los administradores de conexiones de SSIS, especifique la conexión de AWS Glue para el administrador de conexiones de SSIS correspondiente. Asegúrese de que sus conexiones de AWS Glue estén creadas.

   a. En Conexiones, seleccione Conexiones de proyecto.
   b. En Conexión del catálogo de Glue, elija la conexión de AWS Glue correspondiente.
10. Configure el administrador de conexiones de paquetes:

    a. En Conexiones, elija el paquete.
    b. En Conexión del catálogo de Glue, elija la conexión de AWS Glue correspondiente.
    c. Repita estas acciones para todas las conexiones disponibles para el paquete.
11. Seleccione Aplicar.
12. Convierta el paquete. En la vista de árbol de origen, busque Paquetes. Abra el menú contextual del paquete (clic secundario) y elija Convertir paquete.
13. Guarde el script convertido en Amazon S3. En la vista de árbol de destino, busque Scripts del paquete. Abra el menú contextual del paquete (clic secundario) del script convertido y elija Guardar en S3.
14. Configure su trabajo de AWS Glue. En la vista de árbol de destino, busque Scripts del paquete. Abra el menú contextual del paquete (clic secundario) del script convertido y elija Configurar trabajo de AWS Glue.
15. Rellene las secciones de configuración:

    a. Rellene la sección Diseñar flujo de datos:

       - Estrategia de ejecución: elija cómo el trabajo ejecutará los scripts de ETL. Elija SEQUENTIAL para ejecutar los scripts en el orden especificado en el asistente. Elija PARALLEL para ejecutar los scripts en paralelo, sin tener en cuenta el orden especificado en el asistente.
       - Scripts: elija el nombre del script convertido.

- Elija Siguiente.

b. Rellene la sección Propiedades del trabajo:

- Nombre: introduzca el nombre de su trabajo de AWS Glue.
- Rol de IAM: elija el rol de IAM que se utiliza para dar una autorización sobre los recursos que se utilizan para ejecutar el trabajo y obtener acceso a los almacenes de datos.
- Nombre del archivo de script: introduzca el nombre del script convertido.
- Ruta de S3 del archivo de script: introduzca la ruta de Amazon S3 al script convertido.
- Cifrar el script con SSE-S3: elija esta opción para proteger los datos con el cifrado del lado del servidor con claves de cifrado administradas por Amazon S3 (SSE-S3).
- Directorio temporal: introduzca la ruta de Amazon S3 a un directorio temporal para obtener resultados intermedios. las transformaciones integradas AWS Glue y AWS Glue utilizan este directorio para leer o escribir en Amazon Redshift.
- AWS SCT genera automáticamente la ruta para las bibliotecas de Python. Puede revisar esta ruta en la ruta de la biblioteca de Python generada. No puede editar esta ruta generada automáticamente. Para utilizar bibliotecas de Python adicionales, introduzca la ruta en Ruta de la biblioteca de python del usuario.
- Ruta de la biblioteca de python del usuario: introduzca las rutas para bibliotecas de python del usuario opcionales. Separe las rutas de Amazon S3 con comas.
- Ruta de archivos .jar dependientes: introduzca la ruta para los archivos .jar dependientes. Separe las rutas de Amazon S3 con comas.
- Ruta de archivos de referencia: introduzca las rutas de los archivos adicionales, como los archivos de configuración, que requiera el script. Separe las rutas de Amazon S3 con comas.
- Capacidad máxima: introduzca el número máximo de unidades de procesamiento de datos (DPU) de AWS Glue que se pueden asignar cuando se ejecuta este trabajo. Escriba cualquier número entero entre 2 y 100. El valor predeterminado es 2.
- Concurrencia máxima: introduzca el número máximo de ejecuciones concurrentes que están permitidas para este trabajo. El valor predeterminado es 1. AWS Glue devuelve un error cuando se llega a este umbral.
- Tiempo de espera del trabajo (minutos): introduzca el valor del tiempo de espera en su trabajo de ETL como protección contra los trabajos fuera de control. El valor predeterminado es 2 880 minutos (48 horas) para los trabajos por lotes. Cuando el trabajo supera este límite, el estado de ejecución de trabajo cambia a `TIMEOUT`.

- Umbral de notificación de retraso (minutos): introduzca el límite en minutos antes de que AWS SCT envíe una notificación de retraso.
- Número de reintentos: introduzca el número de veces (de 0 a 10) que AWS Glue debe reiniciar automáticamente el trabajo en caso de que se produzca un error. Los trabajos que alcanzan el límite de tiempo de espera no se reinician. El valor predeterminado es 0.
- Elija Siguiente.

c. Configure las conexiones necesarias:

i. En Todas las conexiones, elija las conexiones de AWS Glue necesarias y agréguelas a la lista Conexiones seleccionadas.

ii. Elija Finalizar.

16. Cree un trabajo de AWS Glue configurado. En la vista de árbol de destino, busque y amplíe Trabajos de ETL. Abra el menú contextual (clic secundario) del trabajo de ETL que ha configurado y elija Crear trabajo de AWS Glue.
17. Ejecute el trabajo de AWS Glue:

a. Abra la consola de AWS Glue en https://console.aws.amazon.com/glue/.

b. En el panel de navegación, seleccione Jobs (Trabajos).

c. Seleccione Agregar trabajo y, a continuación, elija el trabajo que desee ejecutar.

d. En el menú Acciones, elija Ejecutar.

Componentes de SSIS que AWS SCT puede convertir a AWS Glue

Puede utilizar AWS SCT para convertir los componentes de flujo de datos y flujo de control, así como los contenedores, los parámetros y las variables.

Entre los componentes de flujo de datos que se admiten se incluyen los siguientes:

- Destino de ADO NET
- Origen de ADO NET
- Agregado
- Transformación de caché
- Transformación de mapas de caracteres
- Transformación dividida condicional
- Transformación de columnas de copia

- Transformación de conversión de datos
- Transformación de columnas derivadas
- Destino de Excel
- Fuente de Excel
- Transformación de columnas de exportación
- Destino de archivo sin formato
- Origen de archivo sin formato
- Transformación de búsquedas difusas
- Transformación de columnas de importación
- Transformación de búsquedas
- Transformación de combinaciones de fusión
- Transformación de fusiones
- Transformación de multidifusiones
- Destino de ODBC
- Origen de ODBC
- Transformación de comandos de OLE DB
- Destino de OLE DB
- Origen de OLE DB
- Transformación de muestreos porcentuales
- Transformación de tablas dinámicas
- Destino de archivo sin procesar
- Origen de archivo sin procesar
- Destino de RecordSet
- Transformación de recuentos de filas
- Transformación de muestreos de filas
- Transformación de datos
- Destino de SQL Server
- Transformación de todas las uniones
- Transformación de tablas no dinámicas
- Origen de XML

Entre los nodos de flujo de control que se admiten se incluyen:

- Tarea de inserción masiva
- Tarea ejecutar paquete
- Tarea ejecutar SQL
- Tarea ejecutar instrucción T-SQL
- Tipo de expresión
- Tarea del sistema de archivos
- Tarea notificar operador
- Tarea enviar correo

Entre los contenedores de SSIS que se admiten se incluyen:

- Para contenedor de bucles
- Contenedor de bucles de Foreach
- Contenedor de secuencias

Convertir SSIS a AWS Glue Studio con AWS SCT

Obtenga información sobre cómo convertir paquetes de Microsoft SQL Server Integration Services (SSIS) a AWS SCT mediante AWS Glue Studio.

Un paquete de SSIS incluye los componentes necesarios, como el administrador de conexiones, las tareas, el flujo de control, el flujo de datos, los parámetros, los controladores de eventos y las variables, para ejecutar una tarea específica de extracción, transformación y carga (ETL). AWS SCT convierte los paquetes de SSIS a un formato compatible con AWS Glue Studio. Tras migrar la base de datos de origen a la Nube de AWS, puede ejecutar estos trabajos de AWS Glue Studio convertidos para realizar tareas de ETL.

Para convertir paquetes de Microsoft SSIS a AWS Glue Studio, utilice la versión 1.0.661 de AWS SCT o una versión posterior.

Temas

- Requisitos previos
- Añadir paquetes de SSIS a su proyecto de AWS SCT

- Conversión de paquetes de SSIS a AWS Glue Studio con AWS SCT
- Crear trabajos de AWS Glue Studio usando el código convertido
- Crear un informe de evaluación para un paquete de SSIS con AWS SCT
- Componentes de SSIS que AWS SCT puede convertir a AWS Glue Studio

Requisitos previos

En esta sección, se explican los requisitos previos para la conversión de paquetes de SSIS a AWS Glue. Estas tareas incluyen la creación de los recursos de AWS necesarios en su cuenta.

Puede usar AWS Identity and Access Management (IAM) para definir políticas y roles que AWS Glue Studio utiliza para acceder a recursos. Para obtener más información, consulte Permisos de IAM para el usuario de AWS Glue Studio.

Una vez que AWS SCT haya convertido los scripts de origen a AWS Glue Studio, cárguelos en un bucket de Amazon S3. Cree este bucket de Amazon S3 y selecciónelo en la configuración del perfil de servicios de AWS. Para obtener más información sobre la creación de un bucket de Amazon S3, consulte Crear su primer bucket de S3 en la Guía del usuario de Amazon Simple Storage Service.

Para asegurarse de que AWS Glue Studio puede conectarse a su almacén de datos, cree un conector personalizado y una conexión. Además, almacene las credenciales de la base de datos en AWS Secrets Manager.

Para crear un conector personalizado

1. Descargue el controlador JDBC para su almacén de datos. Para obtener más información sobre los controladores JDBC que utiliza AWS SCT, consulte. Descarga de los controladores de base de datos necesarios
2. Cargue este archivo de controlador en el bucket de Amazon S3. Para obtener más información, consulte Cargar un objeto en su bucket en la Guía del usuario de Amazon Simple Storage Service.
3. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue Studio en https://console.aws.amazon.com/gluestudio/.
4. Elija Conectores y, a continuación, elija Crear conector personalizado.
5. En URL de S3 del conector, elija Browse S3 y elija el archivo del controlador JDBC que cargó en su bucket de Amazon S3.

6. Escriba un nombre descriptivo para el conector. Por ejemplo, escriba **`SQLServer`**.
7. En Tipo de conector, elija JDBC.
8. En Nombre de clase, introduzca el nombre de la clase principal del controlador JDBC. En Servidor de SQL, introduzca **`com.microsoft.sqlserver.jdbc.SQLServerDriver`**.
9. En Base de URL de JDBC, introduzca la URL base de JDBC. La sintaxis de la URL base de JDBC depende del motor de la base de datos de origen. Para el formato SQL, utilice el siguiente formato: **`jdbc:sqlserver://$<host>:$<port>;databaseName=$<dbname>;user=$<username>;password=$<password>`**.

   Sustituya *`<host>`*, *`<port>`*, *`<dbname>`*, *`<username>`* y *`<password>`* por sus valores.
10. En Delimitador de parámetros de URL, introduzca el punto y coma (`;`).
11. Elija Crear conector.

Para almacenar credenciales de la base de datos en AWS Secrets Manager

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Secrets Manager en https://console.aws.amazon.com/secretsmanager/.
2. Elija Almacenar un secreto nuevo.
3. En la página Elegir tipo de secreto, haga lo siguiente:

   a. En Tipo de secreto, elija Otro tipo de secreto.

   b. En Pares clave/valor, escriba las claves siguientes: **`host`**, **`port`**, **`dbname`**, **`username`** y **`password`**.

   A continuación, introduzca los valores de estas claves.
4. En la página Configurar secreto, introduzca un nombre de secreto descriptivo. Por ejemplo, escriba **`SQL_Server_secret`**.
5. Elija Siguiente. En la página Configurar rotación, seleccione Siguiente.
6. En la página Revisar, revise los detalles del secreto y, a continuación, elija Almacenar.

Para crear una conexión para su conector

1. Inicie sesión en la AWS Management Console y abra la consola de AWS Glue Studio en https://console.aws.amazon.com/gluestudio/.
2. Elija el conector para el que desea crear una conexión y, a continuación, elija Crear conexión.

3. En la página Crear conexión, introduzca un nombre descriptivo para la conexión. Por ejemplo, escriba **`SQL-Server-connection`**.
4. En Secreto de AWS, elija el secreto que creó en AWS Secrets Manager.
5. Configure Opciones de red y, a continuación, seleccione Crear conexión.

Ahora puede crear un trabajo de AWS Glue Studio con un conector personalizado. Para obtener más información, consulte Crear trabajos de AWS Glue Studio.

Añadir paquetes de SSIS a su proyecto de AWS SCT

Puede agregar varios paquetes de SSIS a un único proyecto de AWS SCT.

Para Agregar paquetes de SSIS al proyecto de AWS SCT

1. Cree un proyecto nuevo en AWS SCT o abra un proyecto existente. Para obtener más información, consulte the section called “Creación de un proyecto”.
2. Elija Agregar origen en el menú y, a continuación, elija SQL Server Integration Services.
3. En Nombre de conexión, escriba un nombre para los paquetes de SSIS. AWS SCT muestra este nombre en el árbol del panel izquierdo.
4. En carpeta de paquetes de SSIS, introduzca la ruta a la carpeta con los paquetes de SSIS de origen.
5. Elija Agregar destino en el menú y, a continuación, seleccione AWS Glue Studio.

   Para conectarse a AWS Glue Studio, AWS SCT usa su perfil de AWS. Para obtener más información, consulte Almacenamiento de perfiles de servicio de AWS en AWS SCT.
6. Cree una regla de asignación que incluya su paquete de SSIS de origen y su destino de AWS Glue Studio. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT.
7. Cree conexiones de AWS Glue Studio en la consola de AWS Glue Studio. Para obtener más información, consulte Crear conexiones para conectores.
8. Seleccione Administradores de conexiones en la vista de árbol de la izquierda, abra el menú contextual (clic secundario) y, a continuación, elija Configurar conexiones.

   AWS SCT muestra la ventana Configurar conexiones.
9. Para cada conexión de SSIS de origen, elija una conexión de AWS Glue Studio.

Conversión de paquetes de SSIS a AWS Glue Studio con AWS SCT

A continuación, descubra cómo convertir los paquetes de SSIS a AWS Glue Studio mediante AWS SCT.

Para convertir un paquete de SSIS a AWS Glue Studio

1. Agregue su paquete de SSIS a su proyecto de AWS SCT. Para obtener más información, consulte Añadir paquetes de SSIS a su proyecto de AWS SCT.
2. En el panel izquierdo, expanda los nodos ETL y SSIS.
3. Seleccione Paquetes, abra el menú contextual (clic secundario) y, a continuación, elija Convertir paquete.

   AWS SCT convierte los paquetes de SSIS seleccionados a archivos JSON. Estos objetos JSON representan un nodo en gráficos acíclicos dirigidos (DAG). Busque los archivos convertidos en el nodo DAG de paquete del árbol de la derecha.
4. Elija DAG de paquete, abra el menú contextual (clic secundario) y, a continuación, seleccione Guardar en Amazon S3.

   Ahora puede usar estos scripts para crear trabajos en AWS Glue Studio.

Crear trabajos de AWS Glue Studio usando el código convertido

Después de convertir los paquetes de SSIS de origen, puede usar los archivos JSON convertidos para crear trabajos de AWS Glue Studio.

Para crear un trabajo de AWS Glue Studio

1. Elija DAG de paquete en el árbol derecho, abra el menú contextual (clic secundario) y, a continuación, elija Configurar trabajo de AWS Glue Studio.
2. (Opcional) Aplique el paquete de extensión que emula las funciones del SSIS en AWS Glue Studio.
3. Se abre la ventana Configurar trabajo de AWS Glue Studio.

   Rellene la sección Propiedades básicas del trabajo:

   - Nombre: introduzca el nombre de su trabajo de AWS Glue Studio.
   - Nombre del archivo de script: introduzca el nombre del script de su trabajo.

- Parámetros del trabajo: añada parámetros e introduzca sus valores.

Elija Siguiente.

4. Rellene la sección Propiedades avanzadas del trabajo:

- Rol de IAM: elija el rol de IAM que se utiliza para dar una autorización a AWS Glue Studio y obtener acceso a los almacenes de datos.
- Ruta de S3 del archivo de script: introduzca la ruta de Amazon S3 al script convertido.
- Directorio temporal: introduzca la ruta de Amazon S3 a un directorio temporal para obtener resultados intermedios. AWS Glue Studio utiliza este directorio para leer o escribir en Amazon Redshift.
- AWS SCT genera automáticamente la ruta para las bibliotecas de Python. Puede revisar esta ruta en la ruta de Ruta de la biblioteca de python generada. No puede editar esta ruta generada automáticamente. Para utilizar bibliotecas de Python adicionales, introduzca la ruta en Ruta de la biblioteca de python del usuario.
- Ruta de la biblioteca de python del usuario: introduzca las rutas para bibliotecas de python del usuario opcionales. Separe las rutas de Amazon S3 con comas.
- Ruta de archivos .jar dependientes: introduzca la ruta para los archivos `*.jar` dependientes. Separe las rutas de Amazon S3 con comas.
- Ruta de archivos de referencia: introduzca las rutas de los archivos adicionales, como los archivos de configuración, que requiera el script. Separe las rutas de Amazon S3 con comas.
- Tipo de trabajador: elija `G.1X` o `G.2X`.

  Al elegir `G.1X`, cada trabajador se asigna a 1 DPU (4 vCPU, 16 GB de memoria y disco de 64 GB).

  Al elegir `G.2X`, cada trabajador se asigna a 2 DPU (8 vCPU, 32 GB de memoria y disco de 128 GB).
- Número de trabajadores solicitado: introduzca el número de trabajadores que se asigna cuando se ejecuta el trabajo.
- Concurrencia máx.: introduzca el número máximo de ejecuciones concurrentes que están permitidas para este trabajo. El valor predeterminado es 1. AWS Glue devuelve un error cuando se llega a este umbral.
- Tiempo de espera del trabajo (minutos): introduzca el valor del tiempo de espera en su trabajo de ETL como protección contra los trabajos fuera de control. El valor predeterminado es

2 880 minutos (48 horas) para los trabajos por lotes. Cuando el trabajo supera este límite, el estado de ejecución de trabajo cambia a `TIMEOUT`.

- Umbral de notificación de retraso (minutos): introduzca el límite en minutos antes de que AWS SCT envíe una notificación de retraso.
- Número de reintentos: introduzca el número de veces (de 0 a 10) que AWS Glue debe reiniciar automáticamente el trabajo en caso de que se produzca un error. Los trabajos que alcanzan el límite de tiempo de espera no se reinician. El valor predeterminado es 0.

Elija Finalizar.

AWS SCT configura los trabajos de AWS Glue Studio seleccionados.

5. Busque los trabajos configurados en los trabajos de ETL en el árbol derecho. Elija el trabajo configurado, abra el menú contextual (clic secundario) y elija Crear trabajo de AWS Glue Studio.
6. Elija Aplicar estado y asegúrese de que el valor Estado de su trabajo sea Éxito.
7. Abra la consola de AWS Glue Studio, seleccione Actualizar y elija su trabajo. A continuación, elija Ejecutar.

Crear un informe de evaluación para un paquete de SSIS con AWS SCT

El informe de evaluación de la migración de ETL proporciona información sobre cómo convertir sus paquetes de SSIS a un formato compatible con AWS Glue Studio. El informe de evaluación incluye elementos de acción para los componentes de sus paquetes de SSIS. Estos elementos de acción muestran los componentes que AWS SCT no se puede convertir automáticamente.

Para crear un informe de evaluación de la migración de ETL

1. En el panel izquierdo, expanda el nodo SSIS en ETL.
2. Seleccione Paquetes, abra el menú contextual (clic secundario) y, a continuación, elija Crear informe.
3. Consulte la pestaña Resumen. Aquí, AWS SCT muestra la información del resumen ejecutivo del informe de evaluación de la migración de ETL. Incluye los resultados de conversión de todos los componentes de sus paquetes de SSIS.
4. (Opcional) Guarde una copia local del informe de evaluación de la migración de ETL como archivo PDF o como un archivo de valores separados por comas (CSV):

- Seleccione Guardar en PDF en la parte superior derecha para guardar el informe de evaluación de la migración de ETL como un archivo PDF.

  El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de scripts.
- Seleccione Guardar en CSV en la parte superior derecha para guardar el informe de evaluación de la migración de ETL como un archivo CSV.

  AWS SCT crea tres archivos CSV. Estos archivos contienen elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir los scripts.

5. Haga clic en la pestaña Elementos de acción. Esta pestaña contiene una lista de elementos que requieren una conversión manual a AWS Glue Studio. Si selecciona un elemento de acción de la lista, AWS SCT resalta el elemento de su paquete de SSIS de origen al que se aplica el elemento de acción.

Componentes de SSIS que AWS SCT puede convertir a AWS Glue Studio

Puede utilizar AWS SCT para convertir los componentes y los parámetros del flujo de datos del SSIS a AWS Glue Studio.

Entre los componentes de flujo de datos que se admiten se incluyen los siguientes:

- Destino de ADO NET
- Origen de ADO NET
- Agregado
- Mapa de caracteres
- División condicional
- Columna de copia
- Conversión de datos
- Columna derivada
- Buscar
- Merge
- Merge Join

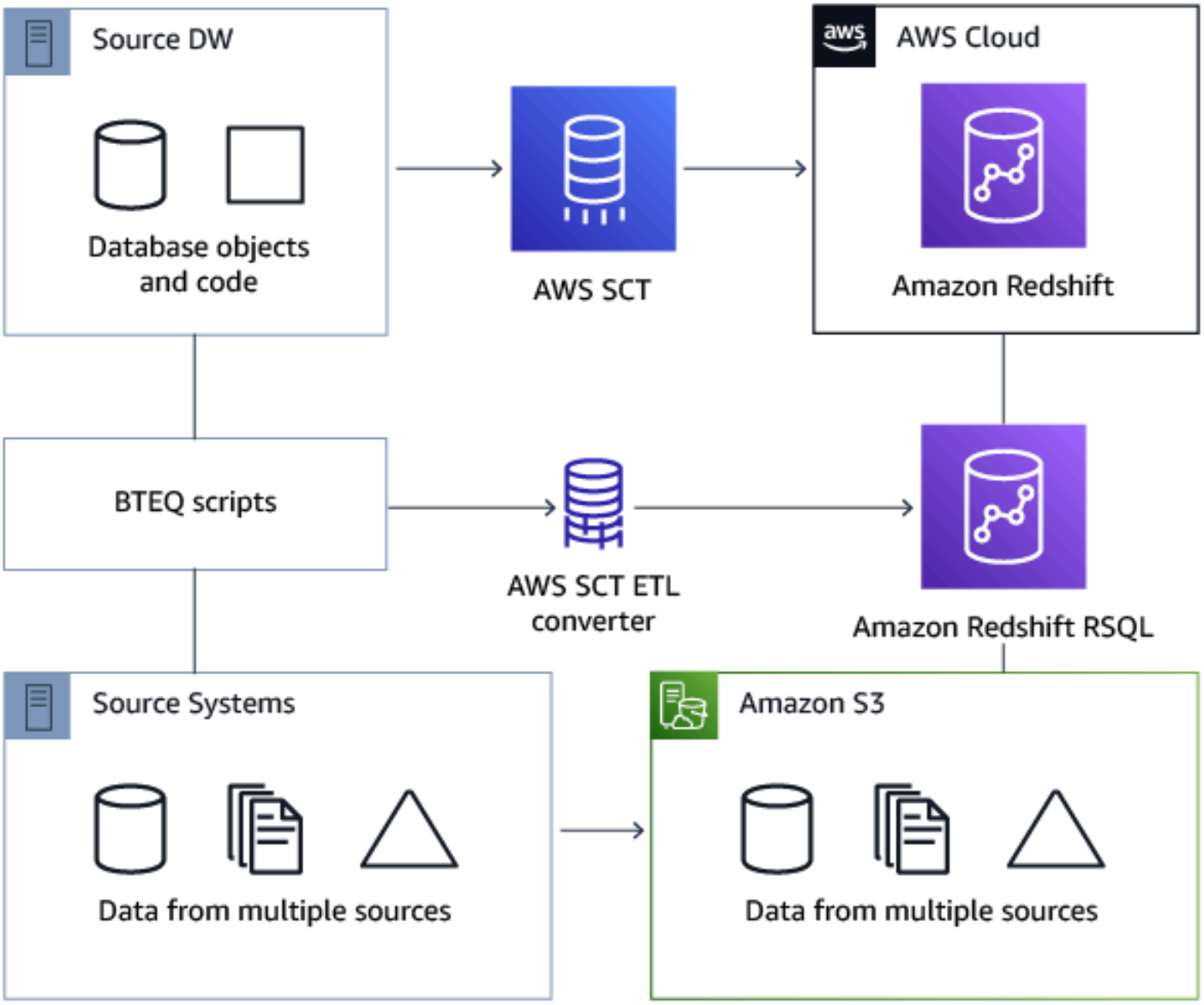
- Multidifusión
- Destino de ODBC
- Origen de ODBC
- Destino de OLEDB
- Origen de OLEDB
- Recuento de filas
- Sort
- Destino de SQL Server
- Union All

AWS SCT puede convertir más componentes de SSIS a AWS Glue. Para obtener más información, consulte Componentes de SSIS que AWS SCT puede convertir a AWS Glue.

Convertir scripts de BTEQ de Teradata a RSQL de Amazon Redshift AWS SCT

Puede usar AWS Schema Conversion Tool (AWS SCT) para convertir scripts de ETL de Basic Teradata Query (BTEQ) de Teradata a RSQL de Amazon Redshift.

El siguiente diagrama de arquitectura muestra un ejemplo de proyecto de migración de bases de datos que incluye la conversión de scripts de extracción, transformación y carga (ETL) a RSQL de Amazon Redshift.



Temas

- Agregar scripts de BTEQ a su proyecto de AWS SCT
- Configurar las variables de sustitución en los scripts de BTEQ con AWS SCT
- Convertir scripts de BTEQ de Teradata a RSQL de Amazon Redshift con AWS SCT
- Administrar scripts de BTEQ con AWS SCT
- Crear un informe de evaluación de la conversión de scripts BTEQ con AWS SCT
- Editar y guardar scripts de BTEQ convertidos con AWS SCT

Agregar scripts de BTEQ a su proyecto de AWS SCT

Puede agregar varios scripts a un único proyecto de AWS SCT.

Para agregar scripts de BTEQ a su proyecto de AWS SCT

1. Cree un proyecto nuevo en AWS SCT o abra un proyecto existente. Para obtener más información, consulte the section called “Creación de un proyecto”.
2. Seleccione Agregar origen en el menú y, a continuación, elija Teradata para agregar la base de datos de origen al proyecto. Para obtener más información, consulte Utilizar Teradata como origen.
3. Seleccione Agregar destino en el menú para agregar una base de datos de Amazon Redshift de destino a su proyecto de AWS SCT.

   Puede utilizar una plataforma de base de datos de destino virtual de Amazon Redshift. Para obtener más información, consulte Usar destinos virtuales.
4. Cree una nueva de asignación nueva que incluya la base de datos Teradata de origen y el destino de Amazon Redshift. Para obtener más información, consulte Cómo agregar una regla de asignación nueva.
5. En el menú Ver, seleccione Vista principal.
6. En el panel izquierdo, expanda el nodo Scripts.
7. Seleccione Scripts de BTEQ, abra el menú contextual (clic secundario) y, a continuación, seleccione Cargar scripts.
8. Introduzca la ubicación del código fuente de sus scripts de BTEQ de Teradata y elija Seleccionar carpeta.

   AWS SCT muestra la ventana Cargar scripts.
9. Haga una de las siguientes acciones:

   a. Si sus scripts de BTEQ de Teradata no incluyen las variables de sustitución, elija Sin variables de sustitución y, a continuación, elija Aceptar para añadir los scripts al proyecto de AWS SCT.

   b. Si sus scripts de BTEQ de Teradata incluyen las variables de sustitución, configure las variables de sustitución. Para obtener más información, consulte Configurar las variables de sustitución en los scripts de BTEQ.

Configurar las variables de sustitución en los scripts de BTEQ con AWS SCT

Sus scripts BTEQ de Teradata pueden incluir variables de sustitución. Por ejemplo, puede utilizar un script de BTEQ con variables de sustitución para ejecutar el mismo conjunto de comandos en varios entornos de bases de datos. Puede utilizar AWS SCT para configurar variables de sustitución en sus scripts de BTEQ.

Antes de ejecutar un script de BTEQ con variables de sustitución, asegúrese de asignar los valores a todas las variables. Para ello, puede utilizar otras herramientas o aplicaciones, como un script de Bash, UC4 (Automic), etc. AWS SCT puede resolver y convertir variables de sustitución solo después de asignar sus valores.

Para configurar variables de sustitución en el script de BTEQ

1. Agregue scripts de BTEQ a su proyecto de AWS SCT. Para obtener más información, consulte Agregar scripts de BTEQ a su proyecto de AWS SCT.

   Cuando agregue sus scripts, elija Se utilizan variables de sustitución.
2. En Definir formato de variable, introduzca una expresión regular que coincida con todas las variables de sustitución del script.

   Por ejemplo, si los nombres de las variables de sustitución comienzan por `${` y terminan por`}`, utilice la expresión regular `\$\{\w+\}`. Para hacer coincidir las variables de sustitución que comienzan por un signo de dólar o un signo de porcentaje, utilice la expresión regular `\$\w+|\%\w+`.

   Las expresiones regulares en AWS SCT cumplen la sintaxis de expresiones regulares de Java. Para obtener más información, consulte Patrón de clase java.uti.regex en la documentación de Java.
3. Seleccione Aceptar para cargar los scripts en su proyecto de AWS SCT y, a continuación, seleccione Aceptar para cerrar la ventana Cargar scripts.
4. Elija Variables para ver todas las variables de sustitución detectadas y sus valores.
5. En Valor, introduzca el valor de la variable de sustitución.

Convertir scripts de BTEQ de Teradata a RSQL de Amazon Redshift con AWS SCT

A continuación, encontrará información sobre cómo convertir scripts de ETL de BTEQ a RSQL de Amazon Redshift mediante AWS SCT.

Para convertir scripts de BTEQ de Teradata a RSQL de Amazon Redshift

1. Agregue sus scripts de BTEQ a su proyecto de AWS SCT. Para obtener más información, consulte Agregar scripts de BTEQ a su proyecto de AWS SCT.
2. Configure las variables de sustitución. Para obtener más información, consulte Configurar las variables de sustitución en los scripts de BTEQ.
3. En el panel izquierdo, expanda el nodo Scripts.
4. Haga una de las siguientes acciones:

   - Para convertir un único script de BTEQ, expanda el nodo Scripts de BTEQ, elija el script que desee convertir y, a continuación, elija Convertir a RSQL en el menú contextual (clic secundario).
   - Para ocultar varios scripts, seleccione todos los scripts que desee convertir. A continuación, elija Scripts de BTEQ, abra el menú contextual (clic secundario) y, a continuación, seleccione Convertir a RSQL en Convertir script.

   AWS SCT convierte todos los scripts de BTEQ de Teradata seleccionados a un formato compatible con RSQL de Amazon Redshift. Busque los scripts convertidos en el nodo Scripts del panel de la base de datos de destino.
5. Edite los scripts de RSQL de Amazon Redshift convertidos o guárdelos. Para obtener más información, consulte Editar y guardar scripts de BTEQ convertidos.

Administrar scripts de BTEQ con AWS SCT

Puede añadir varios scripts de BTEQ o eliminar un script de BTEQ de su proyecto de AWS SCT.

Para agregar un script de BTEQ adicional a su proyecto de AWS SCT

1. En el panel izquierdo, expanda el nodo Scripts.
2. Seleccione el nodo Scripts de BTEQ y abra el menú contextual (clic secundario).

3. Seleccione Cargar scripts.
4. Introduzca la información necesaria para añadir un nuevo script de BTEQ y configurar las variables de sustitución. Para obtener más información, consulte Agregar scripts de BTEQ a su proyecto de AWS SCT y Configurar las variables de sustitución en los scripts de BTEQ.

Para eliminar un script de BTEQ de su proyecto de AWS SCT

1. En el panel izquierdo, expanda el nodo Scripts de BTEQ en Scripts.
2. Elija el script que desea eliminar y abra el menú contextual (clic secundario).
3. Seleccione Eliminar script.

Crear un informe de evaluación de la conversión de scripts BTEQ con AWS SCT

Un informe de evaluación de la conversión de scripts de BTEQ proporciona información sobre la conversión de los comandos y las instrucciones de SQL de BTEQ de sus scripts de BTEQ a un formato compatible con RSQL de Amazon Redshift. El informe de evaluación también incluye elementos de acción para comandos e instrucciones de SQL de BTEQ que AWS SCT no puede convertir.

Para crear un informe de evaluación de la conversión de scripts de BTEQ

1. En el panel izquierdo, expanda el nodo Scripts de BTEQ en Scripts.
2. Elija el script que desea convertir y abra el menú contextual (clic secundario).
3. Seleccione Conversión a RSQL en Crear informe.
4. Consulte la pestaña Resumen. La pestaña Resumen muestra información del resumen ejecutivo del informe de evaluación del script de BTEQ. Incluye los resultados de conversión de todos los comandos y las instrucciones de SQL de BTEQ de sus scripts de BTEQ.
5. (Opcional) Guarde una copia local del informe de evaluación de la conversión de scripts de BTEQ como archivo PDF o como un archivo de valores separados por comas (CSV):

   - Seleccione Guardar en PDF en la parte superior derecha para guardar el informe de evaluación de la conversión de scripts de BTEQ como un archivo PDF.

     El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de scripts.

- Seleccione Guardar en CSV en la parte superior derecha para guardar el informe de evaluación de la conversión de scripts de BTEQ como un archivo CSV.

  El archivo CSV contiene elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir los scripts.

6. Haga clic en la pestaña Elementos de acción. Esta pestaña contiene una lista de elementos que requieren una conversión manual a RSQL de Amazon Redshift. Si selecciona un elemento de acción de la lista, AWS SCT resalta el elemento del script de BTEQ de origen al que se aplica el elemento de acción.

Editar y guardar scripts de BTEQ convertidos con AWS SCT

Puede editar los scripts convertidos en el panel inferior de su proyecto de AWS SCT. AWS SCT almacena el guion editado como parte del proyecto.

Para guardar scripts convertidos

1. Expanda el nodo Scripts de RSQL en Scripts en el panel de la base de datos de destino.
2. Seleccione el script convertido, abra el menú contextual (clic secundario) y elija Guardar Script.
3. Introduzca la ruta a la carpeta para guardar el script convertido y seleccione Guardar.

   AWS SCT guarda el script convertido en un archivo y lo abre.

Conversión de scripts del intérprete de comandos con comandos BTEQ de Teradata incrustados a RSQL de Amazon Redshift con AWS SCT

Puede usar AWS Schema Conversion Tool (AWS SCT) para convertir scripts del intérprete de comandos con comandos Basic Teradata Query (BTEC) de Teradata a scrips del intérprete de comandos con comandos RSQL de Amazon Redshift integrados.

AWS SCT extrae los comandos BTEQ de Teradata de sus scripts del intérprete de comandos y los convierte a un formato compatible con Amazon Redshift. Tras migrar la base de datos de Teradata a Amazon Redshift, puede utilizar estos scripts convertidos para administrar la nueva base de datos de Amazon Redshift.

También puede utilizar AWS SCT para convertir archivos con scripts de ETL de BTEQ de Teradata a RSQL de Amazon Redshift. Para obtener más información, consulte Convertir scripts de BTEQ de Teradata a RSQL de Amazon Redshift AWS SCT.

Temas

- Agregar scripts de intérprete de comandos con comandos de BTEQ de Teradata incrustados a su proyecto de AWS SCT
- Configurar variables de sustitución en scripts del intérprete de comandos con comandos BTEQ de Teradata incrustados con AWS SCT
- Convertir scripts de intérprete de comandos con comandos BTEQ de Teradata integrados con AWS SCT
- Administrar scripts de intérprete de comandos con comandos BTEQ de Teradata integrados con AWS SCT
- Crear un informe de evaluación de la conversión de scripts del intérprete de comandos con AWS SCT
- Editar y guardar scripts del intérprete de comandos convertidos con AWS SCT

Agregar scripts de intérprete de comandos con comandos de BTEQ de Teradata incrustados a su proyecto de AWS SCT

Puede agregar varios scripts a un único proyecto de AWS SCT.

Para agregar un script del intérprete de comandos a su proyecto de AWS SCT

1. Cree un proyecto nuevo en AWS SCT o abra un proyecto existente. Para obtener más información, consulte the section called “Creación de un proyecto”.
2. Seleccione Agregar origen en el menú y, a continuación, elija Teradata para agregar la base de datos de origen al proyecto. Para obtener más información, consulte Utilizar Teradata como origen.
3. Seleccione Agregar destino en el menú para agregar una base de datos de Amazon Redshift de destino a su proyecto de AWS SCT.

   Puede utilizar una plataforma de base de datos de destino virtual de Amazon Redshift. Para obtener más información, consulte Usar destinos virtuales.

4. Cree una nueva de asignación nueva que incluya la base de datos Teradata de origen y el destino de Amazon Redshift. Para obtener más información, consulte Cómo agregar una regla de asignación nueva.
5. En el menú Ver, seleccione Vista principal.
6. En el panel izquierdo, expanda el nodo Scripts.
7. Elija Intérprete de comandos, abra el menú contextual (clic secundario) y elija Cargar scripts.
8. Introduzca la ubicación de los scripts del intérprete de comandos de origen con comandos BTEQ de Teradata incrustados y elija Seleccionar carpeta.

   AWS SCT muestra la ventana Cargar scripts.
9. Haga una de las siguientes acciones:

   - Si sus scripts del intérprete de comandos no incluyen las variables de sustitución, elija Sin variables de sustitución y, a continuación, elija Aceptar para añadir los scripts al proyecto de AWS SCT.
   - Si sus scripts del intérprete de comandos incluyen las variables de sustitución, configure las variables de sustitución. Para obtener más información, consulte Configurar variables de sustitución en scripts de intérprete de comandos.

Configurar variables de sustitución en scripts del intérprete de comandos con comandos BTEQ de Teradata incrustados con AWS SCT

Sus scripts del intérprete de comandos pueden incluir variables de sustitución. Por ejemplo, puede utilizar un único script con variables de sustitución para gestionar bases de datos en distintos entornos. Puede utilizar AWS SCT para configurar variables de sustitución en sus scripts del intérprete de comandos.

Antes de ejecutar los comandos BTEQ con variables de sustitución de un script de intérprete de comandos, asigne los valores a todas las variables de este script de intérprete de comandos. AWS SCT puede resolver y convertir variables de sustitución solo después de asignar sus valores.

Para configurar variables de sustitución en el script del intérprete de comandos

1. Agregue sus scripts del intérprete de comandos de origen a su proyecto de AWS SCT. Para obtener más información, consulte Agregar scripts de intérprete de comandos a su proyecto de AWS SCT.

Cuando agregue sus scripts, elija Se utilizan variables de sustitución.

2. En Definir formato de variable, introduzca una expresión regular que coincida con todas las variables de sustitución del script.

   Por ejemplo, si los nombres de las variables de sustitución comienzan por `${` y terminan por`}`, utilice la expresión regular `\$\{\w+\}`. Para hacer coincidir las variables de sustitución que comienzan por un signo de dólar o un signo de porcentaje, utilice la expresión regular `\$\w+|\%\w+`.

   Las expresiones regulares en AWS SCT cumplen la sintaxis de expresiones regulares de Java. Para obtener más información, consulte Patrón de clase java.uti.regex en la documentación de Java.

3. Seleccione Aceptar para cargar los scripts en su proyecto de AWS SCT y, a continuación, seleccione Aceptar para cerrar la ventana Cargar scripts.
4. Elija Variables para ver todas las variables de sustitución detectadas y sus valores.
5. En Valor, introduzca el valor de la variable de sustitución.

Convertir scripts de intérprete de comandos con comandos BTEQ de Teradata integrados con AWS SCT

A continuación, se explica cómo convertir scripts del intérprete de comandos con comandos BTEQ de Teradata incrustados a RSQL de Amazon Redshift con AWS SCT.

Para convertir un script del intérprete de comandos

1. Agregue sus scripts del intérprete de comandos a su proyecto de AWS SCT. Para obtener más información, consulte Agregar scripts de intérprete de comandos a su proyecto de AWS SCT.
2. Configure las variables de sustitución. Para obtener más información, consulte Configurar variables de sustitución en scripts de intérprete de comandos.
3. En el panel izquierdo, expanda el nodo Scripts.
4. Haga una de las siguientes acciones:

   - Para convertir comandos BTEQ desde un único script del intérprete de comandos, expanda el nodo Intérprete de comandos, elija el script que desee convertir y, a continuación, elija Convertir script en el menú contextual (con el botón derecho).

- Para convertir varios scripts, seleccione todos los scripts que desee convertir. A continuación, seleccione Intérprete de comandos, abra el menú contextual (clic secundario) y, a continuación, seleccione Convertir esquema.

5. Seleccione Aceptar.

   AWS SCT convierte los comandos BTEQ de los scripts del intérprete de comandos seleccionados a un formato compatible con RSQL de Amazon Redshift. Busque los scripts convertidos en el nodo Scripts del panel de la base de datos de destino.

6. Edite los scripts de RSQL de Amazon Redshift convertidos o guárdelos. Para obtener más información, consulte Editar y guardar scripts del intérprete de comandos convertidos.

Administrar scripts de intérprete de comandos con comandos BTEQ de Teradata integrados con AWS SCT

Puede añadir varios scripts del intérprete de comandos o eliminar un script del intérprete de comandos de su proyecto de AWS SCT.

Para agregar un script del intérprete de comandos nuevo a su proyecto de AWS SCT

1. En el panel izquierdo, expanda el nodo Scripts.
2. Seleccione el nodo Intérprete de comandos y abra el menú contextual (clic secundario).
3. Seleccione Cargar scripts.
4. Introduzca la información necesaria para añadir un nuevo script del intérprete de comandos y configurar las variables de sustitución. Para obtener más información, consulte Agregar scripts de intérprete de comandos a su proyecto de AWS SCT y Configurar variables de sustitución en scripts de intérprete de comandos.

Para eliminar un script del intérprete de comandos de su proyecto de AWS SCT

1. Expanda el nodo Intérprete de comandos en Scripts en el panel izquierdo.
2. Elija el script que desea eliminar y abra el menú contextual (clic secundario).
3. Seleccione Eliminar script.

Crear un informe de evaluación de la conversión de scripts del intérprete de comandos con AWS SCT

El informe de evaluación de la conversión de scripts del intérprete de comandos proporciona información sobre la conversión de los comandos y las instrucciones de SQL de BTEQ. La conversión se realiza de los scripts de origen a un formato compatible con RSQL de Amazon Redshift. El informe de evaluación también incluye elementos de acción para comandos e instrucciones de SQL de BTEQ que AWS SCT no puede convertir.

Para crear un informe de evaluación de la conversión de scripts del intérprete de comandos

1. Expanda el nodo Intérprete de comandos en Scripts en el panel izquierdo.
2. Seleccione el script que desea convertir, abra el menú contextual (clic secundario) y, a continuación, seleccione Crear informe.
3. Consulte la pestaña Resumen. La pestaña Resumen muestra información del resumen ejecutivo del informe de evaluación del script del intérprete de comandos. Incluye los resultados de conversión de todos los comandos y las instrucciones de SQL de BTEQ de sus scripts de origen.
4. (Opcional) Guarde una copia local del informe de evaluación de la conversión de scripts del intérprete de comandos como archivo PDF o como un archivo de valores separados por comas (CSV):

   - Seleccione Guardar en PDF en la parte superior derecha para guardar el informe de evaluación de la conversión de scripts del intérprete de comandos como un archivo PDF.

     El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de scripts.
   - Seleccione Guardar en CSV en la parte superior derecha para guardar el informe de evaluación de la conversión de scripts del intérprete de comandos como un archivo CSV.

     El archivo CSV contiene elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir los scripts.
5. Haga clic en la pestaña Elementos de acción. Esta pestaña contiene una lista de elementos que requieren una conversión manual a RSQL de Amazon Redshift. Si selecciona un elemento de acción de la lista, AWS SCT resalta el elemento del script del intérprete de comandos de origen al que se aplica el elemento de acción.

Editar y guardar scripts del intérprete de comandos convertidos con AWS SCT

Puede editar los scripts convertidos en el panel inferior de su proyecto de AWS SCT. AWS SCT almacena el script editado como parte del proyecto.

Para guardar los scripts convertidos

1. Expanda el nodo Scripts de RSQL en Scripts en el panel de la base de datos de destino.
2. Seleccione el script convertido, abra el menú contextual (clic secundario) y elija Guardar Script.
3. Introduzca la ruta a la carpeta para guardar el script convertido y seleccione Guardar.

   AWS SCT guarda el script convertido en un archivo y lo abre.

Convertir scripts de trabajo de FastExport de Teradata a RSQL de Amazon Redshift con AWS SCT

Puede usar AWS Schema Conversion Tool (AWS SCT) para convertir scripts de trabajo de FastExport de Teradata a RSQL de Amazon Redshift.

Un script de trabajo de FastExport es un conjunto de comandos e instrucciones de SQL de FastExport que seleccionan y exportan datos de la base de datos de Teradata. AWS SCT convierte los comandos de FastExport y las instrucciones de SQL a un formato compatible con Amazon Redshift RSQL. Tras migrar la base de datos de Teradata a Amazon Redshift, puede utilizar estos scripts convertidos para exportar datos de la base de datos de Amazon Redshift.

Temas

- Agregar scripts de trabajo de FastExport a su proyecto de AWS SCT
- Configuración de las variables de sustitución en los scripts de trabajo de FastExport de Teradata con AWS SCT
- Convertir scripts de trabajo de FastExport de Teradata con AWS SCT
- Administrar scripts de trabajo de FastExport de Teradata con AWS SCT
- Crear un informe de evaluación de la conversión de scripts de trabajo de FastExport de Teradata con AWS SCT
- Editar y guardar los scripts de trabajo de FastExport de Teradata con AWS SCT

Agregar scripts de trabajo de FastExport a su proyecto de AWS SCT

Puede agregar varios scripts a un único proyecto de AWS SCT.

Para agregar un script de trabajo de FastExport a su proyecto de AWS SCT

1. Cree un proyecto nuevo en AWS SCT o abra un proyecto existente. Para obtener más información, consulte the section called “Creación de un proyecto”.
2. Seleccione Agregar origen en el menú y, a continuación, elija Teradata para agregar la base de datos de origen al proyecto. Para obtener más información, consulte Utilizar Teradata como origen.
3. Seleccione Agregar destino en el menú para agregar una base de datos de Amazon Redshift de destino a su proyecto de AWS SCT.

   Puede utilizar una plataforma de base de datos de destino virtual de Amazon Redshift. Para obtener más información, consulte Usar destinos virtuales.
4. Cree una nueva de asignación nueva que incluya la base de datos Teradata de origen y el destino de Amazon Redshift. Para obtener más información, consulte Cómo agregar una regla de asignación nueva.
5. En el menú Ver, seleccione Vista principal.
6. En el panel izquierdo, expanda el nodo Scripts.
7. Elija FastExport, abra el menú contextual (clic secundario) y elija Cargar scripts.
8. Introduzca la ubicación del código fuente de sus scripts de trabajo de FastExport de Teradata y elija Seleccionar carpeta.

   AWS SCT muestra la ventana Cargar scripts.
9. Haga una de las siguientes acciones:

   - Si sus scripts de trabajo de FastExport de Teradata no incluyen las variables de sustitución, elija Sin variables de sustitución y, a continuación, elija Aceptar para añadir los scripts al proyecto de AWS SCT.
   - Si sus scripts de trabajo de FastExport de Teradata incluyen las variables de sustitución, configure las variables de sustitución. Para obtener más información, consulte Configurar las variables de sustitución en los scripts de trabajo de FastExport de Teradata.

Configuración de las variables de sustitución en los scripts de trabajo de FastExport de Teradata con AWS SCT

Sus scripts de trabajo de FastExport de Teradata pueden incluir variables de sustitución. Por ejemplo, puede utilizar un único script con variables de sustitución para exportar datos de varias bases de datos. Puede utilizar AWS SCT para configurar variables de sustitución en sus scripts de Teradata.

Antes de ejecutar un script de trabajo de FastExport con variables de sustitución, asegúrese de asignar los valores a todas las variables. Para ello, puede utilizar otras herramientas o aplicaciones, como un script de Bash, UC4 (Automic), etc. AWS SCT puede resolver y convertir variables de sustitución solo después de asignar sus valores.

Para configurar variables de sustitución en el script de trabajo de FastExport

1. Agregue sus scripts de trabajo de FastExport de Teradata a su proyecto de AWS SCT. Para obtener más información, consulte Agregar scripts de BTEQ a su proyecto de AWS SCT.

   Cuando agregue sus scripts, elija Se utilizan variables de sustitución.
2. En Definir formato de variable, introduzca una expresión regular que coincida con todas las variables de sustitución del script.

   Por ejemplo, si los nombres de las variables de sustitución comienzan por `${` y terminan por`}`, utilice la expresión regular `\$\{\w+\}`. Para hacer coincidir las variables de sustitución que comienzan por un signo de dólar o un signo de porcentaje, utilice la expresión regular `\$\w+|\%\w+`.

   Las expresiones regulares en AWS SCT cumplen la sintaxis de expresiones regulares de Java. Para obtener más información, consulte Patrón de clase java.uti.regex en la documentación de Java.
3. Seleccione Aceptar para cargar los scripts en su proyecto de AWS SCT y, a continuación, seleccione Aceptar para cerrar la ventana Cargar scripts.
4. En el panel izquierdo, expanda el nodo Scripts. Elija FastExport y, a continuación, elija la carpeta con los scripts. Abra el menú contextual (clic secundario) y elija Exportar variables en Variables de sustitución.
5. Exporte las variables de sustitución para un script. Expanda su carpeta con scripts, elija el script, abra el menú contextual (clic secundario) y elija Exportar variables en Variables de sustitución.

6. Introduzca el nombre del archivo de valores separados por comas (CSV) para guardar las variables de sustitución y elija Guardar.
7. Abra este archivo CSV y rellene los valores de las variables de sustitución.

   Según el sistema operativo, AWS SCT utiliza diferentes formatos para los archivos CSV. Los valores del archivo pueden estar entre comillas o no. Utilice el mismo formato para los valores de las variables de sustitución que para los demás valores del archivo. AWS SCT no puede importar el archivo CSV con valores en diferentes formatos.
8. Guarde el archivo CSV.
9. En el panel izquierdo, expanda el nodo Scripts. Elija FastExport y, a continuación, elija la carpeta con el script. Abra el menú contextual (clic secundario) y elija Exportar variables en Variables de sustitución.
10. Elija el archivo CSV y después elija Abrir.
11. Elija Variables para ver todas las variables de sustitución detectadas y sus valores.

Convertir scripts de trabajo de FastExport de Teradata con AWS SCT

A continuación, se explica cómo convertir scripts de trabajo de FastExport de Teradata a RSQL de Amazon Redshift con AWS SCT.

Para convertir un script de trabajo de FastExport de Teradata a RSQL de Amazon Redshift

1. Agregue sus scripts de trabajo de FastExport a su proyecto de AWS SCT. Para obtener más información, consulte Agregar scripts de trabajo de FastExport a su proyecto de AWS SCT.
2. Configure las variables de sustitución. Para obtener más información, consulte Configurar las variables de sustitución en los scripts de trabajo de FastExport de Teradata.
3. En el panel izquierdo, expanda el nodo Scripts.
4. Haga una de las siguientes acciones:

   - Para convertir un único script de trabajo de FastExport, expanda el nodo FastExport, elija el script que desee convertir y, a continuación, elija Convertir script en el menú contextual (clic secundario).
   - Para convertir varios scripts, seleccione todos los scripts que desee convertir. A continuación, seleccione FastExport, abra el menú contextual (clic secundario) y, a continuación, seleccione Convertir esquema.

AWS SCT convierte todos los scripts de trabajo de FastExport de Teradata seleccionados a un formato compatible con RSQL de Amazon Redshift. Busque los scripts convertidos en el nodo Scripts del panel de la base de datos de destino.

5. Edite los scripts de RSQL de Amazon Redshift convertidos o guárdelos. Para obtener más información, consulte Editar y guardar los scripts de trabajo de FastExport convertidos.

Administrar scripts de trabajo de FastExport de Teradata con AWS SCT

Puede agregar varios scripts de trabajo de FastExport de Teradata o eliminar un script de trabajo de FastExport de su proyecto de AWS SCT.

Para agregar un script de trabajo de FastExport nuevo a su proyecto de AWS SCT

1. En el panel izquierdo, expanda el nodo Scripts.
2. Seleccione el nodo FastExport y abra el menú contextual (clic secundario).
3. Seleccione Cargar scripts.
4. Introduzca la información necesaria para añadir un nuevo script de trabajo de FastExport y configurar las variables de sustitución. Para obtener más información, consulte Agregar scripts de trabajo de FastExport a su proyecto de AWS SCT y Configurar las variables de sustitución en los scripts de trabajo de FastExport de Teradata.

Para eliminar un script de trabajo de FastExport de su proyecto de AWS SCT

1. Expanda el nodo FastExport en Scripts en el panel izquierdo.
2. Elija el script que desea eliminar y abra el menú contextual (clic secundario).
3. Seleccione Eliminar script.

Crear un informe de evaluación de la conversión de scripts de trabajo de FastExport de Teradata con AWS SCT

Un informe de evaluación de la conversión de scripts de trabajo de FastExport proporciona información sobre la conversión de los comandos y las instrucciones de SQL de BTEQ de sus scripts de FastExport a un formato compatible con RSQL de Amazon Redshift. El informe de evaluación

también incluye elementos de acción para comandos e instrucciones de SQL de FastExport que AWS SCT no puede convertir.

Para crear un informe de evaluación de la conversión de scripts de trabajo de FastExport de Teradata

1. Expanda el nodo FastExport en Scripts en el panel izquierdo.
2. Seleccione el script que desea convertir, abra el menú contextual (clic secundario) y, a continuación, seleccione Crear informe.
3. Consulte la pestaña Resumen. La pestaña Resumen muestra información del resumen ejecutivo del informe de evaluación del script de trabajo de FastExport. Incluye los resultados de conversión de todos los comandos y las instrucciones de SQL de FastExport de sus scripts de origen.
4. Puede guardar una copia local del informe de evaluación de la conversión de scripts de trabajo de FastExport en formato PDF o como un archivo de valores separados por comas (CSV).

   a. Seleccione Guardar en PDF en la parte superior derecha para guardar el informe de evaluación de la conversión de scripts de trabajo de FastExport como un archivo PDF.

      El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de scripts.

   b. Seleccione Guardar en CSV en la parte superior derecha para guardar el informe de evaluación de la conversión de scripts de trabajo de FastExport como un archivo CSV.

      El archivo CSV contiene elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir los scripts.
5. Haga clic en la pestaña Elementos de acción. Esta pestaña contiene una lista de elementos que requieren una conversión manual a RSQL de Amazon Redshift. Si selecciona un elemento de acción de la lista, AWS SCT resalta el elemento del script de trabajo de FastExport de origen al que se aplica el elemento de acción.

Editar y guardar los scripts de trabajo de FastExport de Teradata con AWS SCT

Puede editar los scripts convertidos en el panel inferior de su proyecto de AWS SCT. AWS SCT almacena el script editado como parte del proyecto.

Para guardar los scripts convertidos

1. Expanda el nodo Scripts de RSQL en Scripts en el panel de la base de datos de destino.
2. Seleccione el script convertido, abra el menú contextual (clic secundario) y elija Guardar Script.
3. Introduzca la ruta a la carpeta para guardar el script convertido y seleccione Guardar.

   AWS SCT guarda el script convertido en un archivo y lo abre.

Conversión de scripts de trabajo de FastLoad de Teradata a RSQL de Amazon Redshift con AWS SCT

Puede usar AWS Schema Conversion Tool (AWS SCT) para convertir scripts de trabajo de FastLoad de Teradata a RSQL de Amazon Redshift.

Un script de FastLoad de Teradata es un conjunto de comandos que utilizan varias sesiones para cargar datos en una tabla vacía de una base de datos de Teradata. FastLoad de Teradata procesa una serie de comandos e instrucciones de SQL de FastLoad de Teradata. Los comandos FastLoad de Teradata permiten controlar la sesión y gestionar los datos de las transferencias de datos. Las instrucciones de SQL crean, mantienen y eliminan tablas.

AWS SCT convierte los comandos y las instrucciones de SQL de FastLoad de Teradata a un formato compatible con RSQL de Amazon Redshift. Tras migrar la base de datos de Teradata a Amazon Redshift, puede utilizar estos scripts convertidos para cargar datos en la base de datos de Amazon Redshift.

Temas

- Agregar scripts de trabajo de FastLoad a su proyecto de AWS SCT
- Configurar las variables de sustitución en los scripts de trabajo de FastLoad de Teradata con AWS SCT
- Conversión de scripts de trabajo de FastLoad de Teradata con AWS SCT
- Administración de scripts de trabajo de FastLoad de Teradata con AWS SCT
- Crear un informe de evaluación de la conversión de scripts de trabajo de FastLoad de Teradata con AWS SCT
- Editar y guardar los scripts de trabajo de FastLoad de Teradata con AWS SCT

Agregar scripts de trabajo de FastLoad a su proyecto de AWS SCT

Puede agregar varios scripts a un único proyecto de AWS SCT.

Para agregar un script de trabajo de FastLoad a su proyecto de AWS SCT

1. Cree un proyecto nuevo en AWS SCT o abra un proyecto existente. Para obtener más información, consulte the section called “Creación de un proyecto”.
2. Seleccione Agregar origen en el menú y, a continuación, elija Teradata para agregar la base de datos de origen al proyecto. Para obtener más información, consulte Utilizar Teradata como origen.
3. Seleccione Agregar destino en el menú y añada una base de datos de Amazon Redshift de destino a su proyecto de AWS SCT.

   Puede utilizar una plataforma de base de datos de destino virtual de Amazon Redshift. Para obtener más información, consulte Usar destinos virtuales.
4. Cree una nueva de asignación nueva que incluya la base de datos Teradata de origen y el destino de Amazon Redshift. Para obtener más información, consulte Cómo agregar una regla de asignación nueva.
5. En el menú Ver, seleccione Vista principal.
6. En el panel izquierdo, expanda el nodo Scripts.
7. Elija FastLoad, abra el menú contextual (clic secundario) y elija Cargar scripts.
8. Introduzca la ubicación de los scripts de trabajo de FastLoad de Teradata y elija Seleccionar carpeta.

   AWS SCT muestra la ventana Cargar scripts.
9. Haga una de las siguientes acciones:

   - Si sus scripts de trabajo de FastLoad de Teradata no incluyen las variables de sustitución, elija Sin variables de sustitución y, a continuación, elija Aceptar para añadir los scripts al proyecto de AWS SCT.
   - Si sus scripts de trabajo de FastLoad de Teradata incluyen las variables de sustitución, configure las variables de sustitución. Para obtener más información, consulte Configurar las variables de sustitución en los scripts de trabajo de FastLoad de Teradata.

Configurar las variables de sustitución en los scripts de trabajo de FastLoad de Teradata con AWS SCT

Sus scripts de trabajo de FastLoad de Teradata pueden incluir variables de sustitución. Por ejemplo, puede utilizar un único script con variables de sustitución para cargar datos en bases de datos distintas.

Antes de ejecutar un script de trabajo de FastLoad con variables de sustitución, asegúrese de asignar los valores a todas las variables. Para ello, puede utilizar otras herramientas o aplicaciones, como un script de Bash, UC4 (Automic), etc.

AWS SCT puede resolver y convertir variables de sustitución solo después de asignar sus valores. Antes de iniciar la conversión de los scripts de trabajo de FastLoad de Teradata de origen, asigne valores a todas las variables de sustitución. Puede utilizar AWS SCT para configurar variables de sustitución en sus scripts de Teradata.

Para configurar variables de sustitución en el script de trabajo de FastLoad

1. Cuando añada los scripts de trabajo de FastLoad de Teradata de origen a su proyecto de AWS SCT, elija Se utilizan variables de sustitución. Para obtener más información sobre la agregación de estos scripts, consulte Agregar scripts de trabajo de FastLoad a su proyecto de AWS SCT.
2. En Definir formato de variable, introduzca una expresión regular que coincida con todas las variables de sustitución del script.

   Por ejemplo, si los nombres de las variables de sustitución comienzan por `${` y terminan por`}`, utilice la expresión regular `\$\{\w+\}`. Para hacer coincidir las variables de sustitución que comienzan por un signo de dólar o un signo de porcentaje, utilice la expresión regular `\$\w+|\%\w+`.

   Las expresiones regulares en AWS SCT cumplen la sintaxis de expresiones regulares de Java. Para obtener más información, consulte Patrón de clase java.uti.regex en la documentación de Java.
3. Seleccione Aceptar para cargar los scripts en su proyecto de AWS SCT y, a continuación, seleccione Aceptar para cerrar la ventana Cargar scripts.
4. En el panel izquierdo, expanda el nodo Scripts. Elija FastLoad y, a continuación, elija la carpeta con los scripts. Abra el menú contextual (clic secundario) y elija Exportar variables en Variables de sustitución.

Además, puede exportar las variables de sustitución para un script. Expanda su carpeta con scripts, elija el script, abra el menú contextual (clic secundario) y elija Exportar variables en Variables de sustitución.

5. Introduzca el nombre del archivo de valores separados por comas (CSV) para guardar las variables de sustitución y elija Guardar.
6. Abra este archivo CSV y rellene los valores de las variables de sustitución.

   Según el sistema operativo, AWS SCT utiliza diferentes formatos para el archivos CSV. Los valores del archivo pueden estar entre comillas o no. Utilice el mismo formato para los valores de las variables de sustitución que para los demás valores del archivo. AWS SCT no puede importar el archivo CSV con valores en diferentes formatos.
7. Guarde el archivo CSV.
8. En el panel izquierdo, expanda el nodo Scripts. Elija FastLoad y, a continuación, elija la carpeta con el script. Abra el menú contextual (clic secundario) y elija Exportar variables en Variables de sustitución.
9. Elija el archivo CSV y luego elija Abrir.
10. Elija Variables para ver todas las variables de sustitución detectadas y sus valores.

Conversión de scripts de trabajo de FastLoad de Teradata con AWS SCT

A continuación, se explica cómo convertir scripts de trabajo de FastLoad de Teradata a RSQL de Amazon Redshift con AWS SCT.

Para convertir un script de trabajo de FastLoad de Teradata a RSQL de Amazon Redshift

1. Agregue sus scripts de trabajo de FastLoad a su proyecto de AWS SCT. Para obtener más información, consulte Cómo agregar scripts de trabajo de FastLoad al proyecto de AWS SCT.
2. Configure las variables de sustitución. Para obtener más información, consulte Configurar las variables de sustitución en los scripts de trabajo de FastLoad de Teradata.
3. En el panel izquierdo, expanda el nodo Scripts.
4. Haga una de las siguientes acciones:

   - Para convertir un único script de trabajo de FastLoad, expanda el nodo FastLoad, elija el script que desee convertir y, a continuación, elija Convertir script en el menú contextual (clic secundario).

- Para convertir varios scripts, seleccione todos los scripts que desee convertir. Seleccione FastExport, abra el menú contextual (clic secundario) y, a continuación, seleccione Convertir script. A continuación, realice alguna de las operaciones siguientes:
  - Si almacena el archivo de datos de origen en Amazon S3, elija Ruta del objeto de S3 como Ubicación del archivo de datos de origen.

    Introduzca los valores en Carpeta de bucket de Amazon S3 y en Bucket de Amazon S3 para el archivo de manifiesto del archivo de datos de origen.
  - Si almacena el archivo de datos de origen en Amazon S3, elija Dirección de host como Ubicación del archivo de datos de origen.

    Introduzca los valores en URL o dirección IP del host, Nombre de usuario del host y Bucket de Amazon S3 para el archivo de manifiesto del archivo de datos de origen.

5. Seleccione Aceptar.

   AWS SCT convierte todos los scripts de trabajo de FastLoad de Teradata seleccionados a un formato compatible con RSQL de Amazon Redshift. Busque los scripts convertidos en el nodo Scripts del panel de la base de datos de destino.
6. Edite los scripts de RSQL de Amazon Redshift convertidos o guárdelos. Para obtener más información, consulte Editar y guardar los scripts de trabajo de FastLoad convertidos.

Administración de scripts de trabajo de FastLoad de Teradata con AWS SCT

Puede agregar varios scripts de trabajo de FastLoad de Teradata o eliminar un script de trabajo de FastLoad de su proyecto de AWS SCT.

Para agregar un script de trabajo de FastLoad nuevo a su proyecto de AWS SCT

1. En el panel izquierdo, expanda el nodo Scripts.
2. Seleccione el nodo FastLoad y abra el menú contextual (clic secundario).
3. Seleccione Cargar scripts.
4. Introduzca la información necesaria para añadir un nuevo script de trabajo de FastLoad y configurar las variables de sustitución. Para obtener más información, consulte Cómo agregar scripts de trabajo de FastLoad al proyecto de AWS SCT y Configurar las variables de sustitución en los scripts de trabajo de FastLoad de Teradata.

Para eliminar un script de trabajo de FastLoad de su proyecto de AWS SCT

1. Expanda el nodo FastLoad en Scripts en el panel izquierdo.
2. Elija el script que desea eliminar y abra el menú contextual (clic secundario).
3. Seleccione Eliminar script.

Crear un informe de evaluación de la conversión de scripts de trabajo de FastLoad de Teradata con AWS SCT

El informe de evaluación de la conversión de scripts de trabajo de FastLoad proporciona información sobre la conversión de los comandos y las instrucciones de SQL de FastLoad. La conversión se realiza de los scripts de origen a un formato compatible con RSQL de Amazon Redshift. El informe de evaluación también incluye elementos de acción para comandos e instrucciones de SQL de FastLoad que AWS SCT no puede convertir.

Para crear un informe de evaluación de la conversión de scripts de trabajo de FastLoad de Teradata

1. Expanda el nodo FastLoad en Scripts en el panel izquierdo.
2. Seleccione el script que desea convertir, abra el menú contextual (clic secundario) y, a continuación, seleccione Crear informe.
3. Consulte la pestaña Resumen.

   La pestaña Resumen muestra información del resumen ejecutivo del informe de evaluación del script de trabajo de FastLoad. Incluye los resultados de conversión de todos los comandos y las instrucciones de SQL de FastLoad de sus scripts de origen.
4. (Opcional) Guarde una copia local del informe de evaluación de la conversión de scripts de trabajo de FastLoad como archivo PDF o como un archivo de valores separados por comas (CSV):

   - Seleccione Guardar en PDF en la parte superior derecha para guardar el informe de evaluación de la conversión de scripts de trabajo de FastLoad como un archivo PDF.

     El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de scripts.
   - Seleccione Guardar en CSV en la parte superior derecha para guardar el informe de evaluación de la conversión de scripts de trabajo de FastLoad como un archivo CSV.

El archivo CSV contiene elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir los scripts.

5. Haga clic en la pestaña Elementos de acción. Esta pestaña contiene una lista de elementos que requieren una conversión manual a RSQL de Amazon Redshift. Si selecciona un elemento de acción de la lista, AWS SCT resalta el elemento del script de trabajo de FastLoad de origen al que se aplica el elemento de acción.

Editar y guardar los scripts de trabajo de FastLoad de Teradata con AWS SCT

Puede editar los scripts convertidos en el panel inferior de su proyecto de AWS SCT. AWS SCT almacena el script editado como parte del proyecto.

Para guardar los scripts convertidos

1. Expanda el nodo Scripts de RSQL en Scripts en el panel de la base de datos de destino.
2. Seleccione el script convertido, abra el menú contextual (clic secundario) y elija Guardar Script.
3. Introduzca la ruta a la carpeta para guardar el script convertido y seleccione Guardar.

   AWS SCT guarda el script convertido en un archivo y lo abre.

Convertir scripts de trabajo de MultiLoad de Teradata a RSQL de Amazon Redshift con AWS SCT

Puede usar AWS SCT para convertir scripts de trabajo de MultiLoad de Teradata a RSQL de Amazon Redshift.

Un script de trabajo de MultiLoad de Teradata es un conjunto de comandos para el mantenimiento por lotes de su base de datos de Teradata. Una tarea de importación de MultiLoad de Teradata lleva a cabo distintas operaciones de inserción, actualización y eliminación en hasta cinco tablas y vistas diferentes. Las tareas de eliminación de MultiLoad de Teradata pueden eliminar un gran número de filas de una sola tabla.

AWS SCT convierte los comandos y las instrucciones de SQL de MultiLoad de Teradata a un formato compatible con RSQL de Amazon Redshift. Tras migrar la base de datos de Teradata a Amazon

Redshift, puede utilizar estos scripts convertidos para administrar los datos de su base de datos de Amazon Redshift.

Temas

- Cómo agregar scripts de trabajo de MultiLoad a su proyecto de AWS SCT
- Configurar las variables de sustitución en los scripts de trabajo de MultiLoad de Teradata con AWS SCT
- Conversión de scripts de trabajo de MultiLoad de Teradata con AWS SCT
- Administración d scripts de trabajo de MultiLoad de Teradata con AWS SCT
- Crear un informe de evaluación de la conversión de scripts de trabajo de MultiLoad de Teradata con AWS SCT
- Editar y guardar los scripts de trabajo de MultiLoad de Teradata con AWS SCT

Cómo agregar scripts de trabajo de MultiLoad a su proyecto de AWS SCT

Puede agregar varios scripts a un único proyecto de AWS SCT.

Para agregar un script de trabajo de MultiLoad a su proyecto de AWS SCT

1. Cree un proyecto nuevo en AWS SCT o abra un proyecto existente. Para obtener más información, consulte the section called “Creación de un proyecto”.
2. Seleccione Agregar origen en el menú y, a continuación, elija Teradata para agregar la base de datos de origen al proyecto. Para obtener más información, consulte Utilizar Teradata como origen.
3. Seleccione Agregar destino en el menú para agregar una base de datos de Amazon Redshift de destino a su proyecto de AWS SCT.

   Puede utilizar una plataforma de base de datos de destino virtual de Amazon Redshift. Para obtener más información, consulte Usar destinos virtuales.
4. Cree una nueva de asignación nueva que incluya la base de datos Teradata de origen y el destino de Amazon Redshift. Para obtener más información, consulte Cómo agregar una regla de asignación nueva.
5. En el menú Ver, seleccione Vista principal.
6. En el panel izquierdo, expanda el nodo Scripts.
7. Elija MultiLoad, abra el menú contextual (clic secundario) y elija Cargar scripts.

8. Introduzca la ubicación de los scripts de trabajo de MultiLoad de Teradata y elija Seleccionar carpeta.

   AWS SCT muestra la ventana Cargar scripts.

9. Haga una de las siguientes acciones:

   - Si sus scripts de trabajo de MultiLoad de Teradata no incluyen las variables de sustitución, elija Sin variables de sustitución y, a continuación, elija Aceptar para añadir los scripts al proyecto de AWS SCT.
   - Si sus scripts de trabajo de MultiLoad de Teradata incluyen las variables de sustitución, configure las variables de sustitución. Para obtener más información, consulte Configurar las variables de sustitución en los scripts de trabajo de MultiLoad.

Configurar las variables de sustitución en los scripts de trabajo de MultiLoad de Teradata con AWS SCT

Sus scripts de trabajo de MultiLoad de Teradata pueden incluir variables de sustitución. Por ejemplo, puede utilizar un único script con variables de sustitución para cargar datos en bases de datos distintas.

Antes de ejecutar un script de trabajo de MultiLoad con variables de sustitución, asegúrese de asignar los valores a todas las variables. Para ello, puede utilizar otras herramientas o aplicaciones, como un script de Bash, UC4 (Automic), etc.

AWS SCT puede resolver y convertir variables de sustitución solo después de asignar sus valores. Antes de iniciar la conversión de los scripts de trabajo de MultiLoad de Teradata de origen, asigne valores a todas las variables de sustitución. Puede utilizar AWS SCT para configurar variables de sustitución en sus scripts de Teradata.

Para configurar variables de sustitución en el script de trabajo de MultiLoad

1. Cuando añada los scripts de trabajo de MultiLoad de Teradata de origen a su proyecto de AWS SCT, elija Se utilizan variables de sustitución. Para obtener más información sobre la agregación de estos scripts, consulte Cómo agregar scripts de trabajo de MultiLoad a su proyecto de AWS SCT.
2. En Definir formato de variable, introduzca una expresión regular que coincida con todas las variables de sustitución del script.

Por ejemplo, si los nombres de las variables de sustitución comienzan por `${` y terminan por`}`, utilice la expresión regular `\$\{\w+\}`. Para hacer coincidir las variables de sustitución que comienzan por un signo de dólar o un signo de porcentaje, utilice la expresión regular `\$\w+|\%\w+`.

Las expresiones regulares en AWS SCT cumplen la sintaxis de expresiones regulares de Java. Para obtener más información, consulte Patrón de clase java.uti.regex en la documentación de Java.

3. Seleccione Aceptar para cargar los scripts en su proyecto de AWS SCT y, a continuación, seleccione Aceptar para cerrar la ventana Cargar scripts.
4. Elija Variables para ver todas las variables de sustitución detectadas y sus valores.
5. En Valor, introduzca el valor de la variable de sustitución.

Conversión de scripts de trabajo de MultiLoad de Teradata con AWS SCT

A continuación, se explica cómo convertir scripts de trabajo de MultiLoad de Teradata a RSQL de Amazon Redshift con AWS SCT.

Puede convertir scripts de trabajo de MultiLoad de Teradata a RSQL de Amazon Redshift

1. Agregue sus scripts de trabajo de MultiLoad a su proyecto de AWS SCT. Para obtener más información, consulte Cómo agregar scripts de trabajo de MultiLoad a su proyecto de AWS SCT.
2. Configure las variables de sustitución e introduzca sus valores. Para obtener más información, consulte Configurar las variables de sustitución en los scripts de trabajo de MultiLoad.
3. En el panel izquierdo, expanda el nodo Scripts.
4. Haga una de las siguientes acciones:

   - Para convertir un único script de trabajo de MultiLoad, expanda el nodo MultiLoad, elija el script que desee convertir y, a continuación, elija Convertir script en el menú contextual (clic secundario).
   - Para convertir varios scripts, seleccione todos los scripts que desee convertir. Elija MultiLoad, abra el menú contextual (clic secundario) y elija Convertir script.
5. Haga una de las siguientes acciones:

   - Si almacena el archivo de datos de origen en Amazon S3, elija Ruta del objeto de S3 como Ubicación del archivo de datos de origen.

Introduzca los valores en Carpeta de bucket de Amazon S3 y en Bucket de Amazon S3 para el archivo de manifiesto del archivo de datos de origen.

- Si almacena el archivo de datos de origen en Amazon S3, elija Dirección de host como Ubicación del archivo de datos de origen.

  Introduzca los valores en URL o dirección IP del host, Nombre de usuario del host y Bucket de Amazon S3 para el archivo de manifiesto del archivo de datos de origen.

6. Seleccione Aceptar.

   AWS SCT convierte todos los scripts de trabajo de MultiLoad de Teradata seleccionados a un formato compatible con RSQL de Amazon Redshift. Busque los scripts convertidos en el nodo Scripts del panel de la base de datos de destino.

7. Edite los scripts de RSQL de Amazon Redshift convertidos o guárdelos. Para obtener más información, consulte Editar y guardar los scripts de trabajo de MultiLoad convertidos.

Administración d scripts de trabajo de MultiLoad de Teradata con AWS SCT

Puede agregar varios scripts de trabajo de MultiLoad de Teradata o eliminar un script de trabajo de MultiLoad de su proyecto de AWS SCT.

Para agregar un script de trabajo de MultiLoad nuevo a su proyecto de AWS SCT

1. En el panel izquierdo, expanda el nodo Scripts.
2. Seleccione el nodo MultiLoad y abra el menú contextual (clic secundario).
3. Seleccione Cargar scripts.
4. Introduzca la información necesaria para añadir un nuevo script de trabajo de MultiLoad y configurar las variables de sustitución. Para obtener más información, consulte Cómo agregar scripts de trabajo de MultiLoad a su proyecto de AWS SCT y Configurar las variables de sustitución en los scripts de trabajo de MultiLoad.

Para eliminar un script de trabajo de MultiLoad de su proyecto de AWS SCT

1. Expanda el nodo MultiLoad en Scripts en el panel izquierdo.
2. Elija el script que desea eliminar y abra el menú contextual (clic secundario).
3. Seleccione Eliminar script.

Crear un informe de evaluación de la conversión de scripts de trabajo de MultiLoad de Teradata con AWS SCT

El informe de evaluación de la conversión de scripts de trabajo de MultiLoad proporciona información sobre la conversión de los comandos y las instrucciones de SQL de MultiLoad. La conversión se realiza de los scripts de origen a comandos e instrucciones de SQL de Amazon Redshift para Amazon Redshift. El informe de evaluación también incluye elementos de acción para comandos e instrucciones de SQL de MultiLoad que AWS SCT no puede convertir.

Para crear un informe de evaluación de la conversión de scripts de trabajo de MultiLoad de Teradata

1. Expanda el nodo MultiLoad en Scripts en el panel izquierdo.
2. Elija los scripts para los que desee crear el informe de evaluación, abra el menú contextual (clic secundario) y, a continuación, elija Crear informe.
3. Consulte la pestaña Resumen. La pestaña Resumen muestra información del resumen ejecutivo del informe de evaluación del script de trabajo de MultiLoad. Incluye los resultados de conversión de todos los comandos y las instrucciones de SQL de MultiLoad de sus scripts de origen.
4. (Opcional) Guarde una copia local del informe de evaluación de la conversión de scripts de trabajo de MultiLoad como archivo PDF o como archivos de valores separados por comas (CSV):

   - Seleccione Guardar en PDF en la parte superior derecha para guardar el informe de evaluación de la conversión de scripts de trabajo de MultiLoad como un archivo PDF.

     El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de scripts.
   - Seleccione Guardar en CSV en la parte superior derecha para guardar el informe de evaluación de la conversión de scripts de trabajo de MultiLoad como un archivo CSV.

     AWS SCT crea dos archivos CSV. Estos archivos contienen el resumen ejecutivo, elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir los scripts.
5. Haga clic en la pestaña Elementos de acción. Esta pestaña contiene una lista de elementos que requieren una conversión manual a RSQL de Amazon Redshift. Si selecciona un elemento de acción de la lista, AWS SCT resalta el elemento del script de trabajo de MultiLoad de origen al que se aplica el elemento de acción.

Editar y guardar los scripts de trabajo de MultiLoad de Teradata con AWS SCT

Puede editar los scripts convertidos en el panel inferior de su proyecto de AWS SCT. AWS SCT almacena el script editado como parte del proyecto.

Para guardar los scripts convertidos

1. Expanda el nodo Scripts de RSQL en Scripts en el panel de la base de datos de destino.
2. Seleccione el script convertido, abra el menú contextual (clic secundario) y elija Guardar Script.
3. Introduzca la ruta a la carpeta para guardar el script convertido y seleccione Guardar.

   AWS SCT guarda el script convertido en un archivo y lo abre.

Migración de marcos de macrodatos con AWS Schema Conversion Tool

Puede usar AWS Schema Conversion Tool (AWS SCT) para migrar marcos de macrodatos a la Nube de AWS.

Actualmente, AWS SCT admite la migración de clústeres de Hadoop a Amazon EMR y Amazon S3. Este proceso de migración incluye los servicios de Hive y HDFS.

Además, puede utilizar AWS SCT para automatizar la conversión de los flujos de trabajo de orquestación de Apache Oozie a AWS Step Functions.

Temas

- Migración de Apache Hadoop a Amazon EMR con AWS Schema Conversion Tool
- Conversión de Apache Oozie a AWS Step Functions con AWS Schema Conversion Tool

Migración de Apache Hadoop a Amazon EMR con AWS Schema Conversion Tool

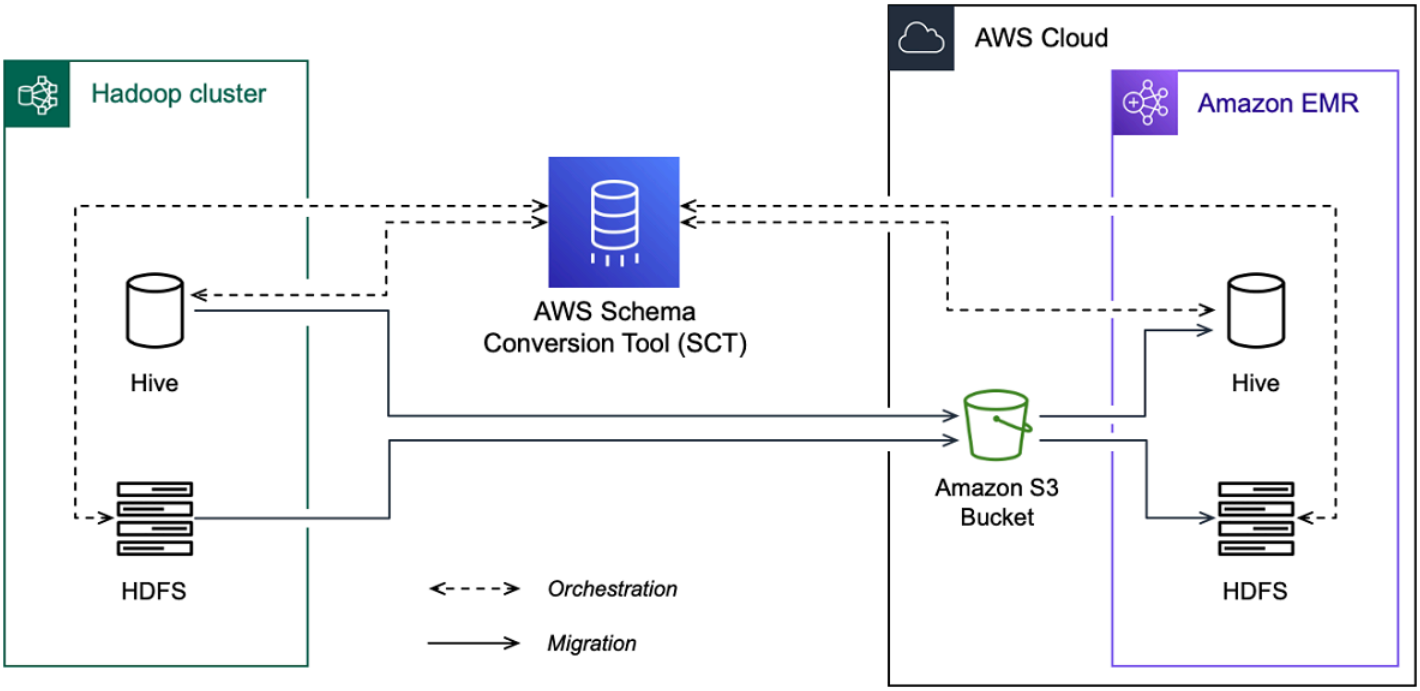
Para migrar los clústeres de Apache Hadoop, utilice la versión 1.0.670 o posterior de AWS SCT. Además, familiarícese con la interfaz de la línea de comandos (CLI) de AWS SCT. Para obtener más información, consulte AWS SCT Referencia CLI.

Temas

- Información general sobre la migración
- Paso 1: Conectarse a los clústeres de Hadoop
- Paso 2: Configurar las reglas de asignación
- Paso 3: Crear un informe de evaluación
- Paso 4: Migrar el clúster de Apache Hadoop a Amazon EMR con AWS SCT
- Ejecución del script de la CLI
- Administrar el proyecto de migración de macrodatos

Información general sobre la migración

La siguiente imagen muestra el diagrama de la arquitectura de la migración de Apache Hadoop a Amazon EMR.



AWS SCT migra datos y metadatos del clúster de Hadoop de origen a un bucket de Amazon S3. A continuación, AWS SCT utiliza los metadatos de Hive de origen para crear objetos de base de datos en el servicio de Hive de Amazon EMR. Si lo desea, puede configurar Hive para que AWS Glue Data Catalog lo utilice como metaalmacén. En este caso, AWS SCT migra los metadatos de Hive de origen a AWS Glue Data Catalog.

A continuación, puede utilizar AWS SCT para migrar los datos de un bucket de Amazon S3 al servicio Amazon EMR HDFS de destino. Como alternativa, puede dejar los datos en el bucket de Amazon S3 y utilizarlos como repositorio de datos para las cargas de trabajo de Hadoop.

Para iniciar la migración de Hapood, debe crear y ejecutar el script de la CLI de AWS SCT. Este script incluye el conjunto completo de comandos para ejecutar la migración. Puede descargar y editar una plantilla del script de migración de Hadoop. Para obtener más información, consulte Obtención de escenarios de la CLI.

Asegúrese de que el script incluye los siguientes pasos para poder ejecutar la migración de Apache Hadoop a Amazon S3 y Amazon EMR.

Paso 1: Conectarse a los clústeres de Hadoop

Para iniciar la migración del clúster de Apache Hadoop, cree un proyecto de AWS SCT nuevo. A continuación, conéctese a los clústeres de origen y destino. Cree y aprovisione los recursos de AWS de destino antes de iniciar la migración.

En este paso, utilice los siguientes comandos de la CLI de AWS SCT.

- `CreateProject`: para crear un proyecto de AWS SCT nuevo.
- `AddSourceCluster`: para conectarse al clúster de Hadoop de origen del proyecto de AWS SCT.
- `AddSourceClusterHive`: para conectarse al servicio de Hive de origen del proyecto.
- `AddSourceClusterHDFS`: para conectarse al servicio de HDFS de origen del proyecto.
- `AddTargetCluster`: para conectarse al clúster de Amazon EMR de destino del proyecto.
- `AddTargetClusterS3`: para agregar el bucket de Amazon S3 al proyecto.
- `AddTargetClusterHive`: para conectarse al servicio de Hive de destino del proyecto
- `AddTargetClusterHDFS`: para conectarse al servicio de HDFS de destino del proyecto

Para obtener ejemplos de uso de estos comandos de la CLI de AWS SCT, consulte Usar Apache Hadoop como origen.

Cuando ejecuta el comando que conecta a un clúster de origen o destino, AWS SCT intenta establecer la conexión con este clúster. Si el intento de conexión falla, AWS SCT deja de ejecutar los comandos del script de la CLI y muestra un mensaje de error.

Paso 2: Configurar las reglas de asignación

Después de conectarse a los clústeres de origen y destino, configure las reglas de asignación. Una regla de asignación define el objetivo de migración de un clúster de origen. Configure reglas de asignación para todos los clústeres de origen que haya agregado al proyecto de AWS SCT. Para obtener más información acerca de las reglas de asignación, consulte Creación de reglas de asignación en AWS SCT.

En este paso, utilice el comando `AddServerMapping`. Este comando usa dos parámetros, que definen los clústeres de origen y destino. Puede utilizar el comando `AddServerMapping` con la ruta explícita a los objetos de la base de datos o con el nombre de un objeto. En la primera opción, debe incluir el tipo de objeto y su nombre. Para la segunda opción, incluya solo los nombres de los objetos.

- `sourceTreePath`: la ruta explícita a los objetos de la base de datos de origen.

  `targetTreePath`: la ruta explícita a los objetos de la base de datos de destino.
- `sourceNamePath`: la ruta que incluye solo los nombres de los objetos de origen.

  `targetNamePath`: la ruta que incluye solo los nombres de los objetos de destino.

El siguiente ejemplo de código crea una regla de asignación mediante rutas explícitas para la base de datos de Hive `testdb` de origen y el clúster de EMR de destino.

```
AddServerMapping
 -sourceTreePath: 'Clusters.HADOOP_SOURCE.HIVE_SOURCE.Databases.testdb'
 -targetTreePath: 'Clusters.HADOOP_TARGET.HIVE_TARGET'
/
```

Puede usar este ejemplo y los siguientes en Windows. Para ejecutar los comandos de la CLI en Linux, actualice las rutas de los archivos de forma adecuada según el sistema operativo.

El siguiente ejemplo de código crea una regla de asignación utilizando las rutas que incluyen solo los nombres de los objetos.

```
AddServerMapping
 -sourceNamePath: 'HADOOP_SOURCE.HIVE_SOURCE.testdb'
 -targetNamePath: 'HADOOP_TARGET.HIVE_TARGET'
/
```

Puede elegir Amazon EMR o Amazon S3 como destino para el objeto de origen. Para cada objeto de origen, solo puede elegir un destino en un único proyecto de AWS SCT. Para cambiar el destino de migración de un objeto de origen, elimine la regla de asignación existente y, a continuación, cree una regla de asignación nueva. Para eliminar una regla de asignación, utilice el comando `DeleteServerMapping`. Este comando usa uno de los dos parámetros siguientes.

- `sourceTreePath`: la ruta explícita a los objetos de la base de datos de origen.
- `sourceNamePath`: la ruta que incluye solo los nombres de los objetos de origen.

Para obtener más información sobre los comandos `AddServerMapping` y `DeleteServerMapping`, consulte la Referencia de la CLI de AWS Schema Conversion Tool.

Paso 3: Crear un informe de evaluación

Antes de iniciar la migración, le recomendamos que cree un informe de evaluación. Este informe resume todas las tareas de migración y detalla las acciones que surgirán durante la migración. Para asegurarse de que la migración no falle, consulte este informe y aborde las medidas que se deben tomar antes de la migración. Para obtener más información, consulte Informes de evaluación de migración.

En este paso, utilice el comando `CreateMigrationReport`. Este comando emplea dos parámetros. El parámetro `treePath` es obligatorio y el parámetro `forceMigrate` es opcional.

- `treePath`: la ruta explícita a los objetos de la base de datos de origen para la que guarda una copia del informe de evaluación.
- `forceMigrate`: si se establece en `true`, AWS SCT continúa la migración aunque el proyecto incluya una carpeta de HDFS y una tabla de Hive que hagan referencia al mismo objeto. El valor predeterminado es `false`.

Puede guardar una copia del informe de evaluación en formato PDF o como un archivo de valores separados por comas (CSV). Para ello, utilice el comando `SaveReportPDF` o `SaveReportCSV`.

El comando `SaveReportPDF` guarda una copia del informe de evaluación en un archivo PDF. Este comando utiliza cuatro parámetros. El parámetro `file` es obligatorio, mientras que otros parámetros son opcionales.

- `file`: la ruta al archivo PDF y su nombre.
- `filter`: el nombre del filtro que creó anteriormente para definir el alcance de los objetos de origen que desee migrar.
- `treePath`: la ruta explícita a los objetos de la base de datos de origen para los que guarda una copia del informe de evaluación.
- `namePath`: la ruta que incluye solo los nombres de los objetos de destino para los que guarda una copia del informe de evaluación.

El comando `SaveReportCSV` guarda el informe de evaluación en tres archivos CSV. Este comando utiliza cuatro parámetros. El parámetro `directory` es obligatorio, mientras que otros parámetros son opcionales.

- `directory`: la ruta a la carpeta en la que AWS SCT guarda los archivos CSV.

- `filter`: el nombre del filtro que creó anteriormente para definir el alcance de los objetos de origen que desee migrar.
- `treePath`: la ruta explícita a los objetos de la base de datos de origen para los que guarda una copia del informe de evaluación.
- `namePath`: la ruta que incluye solo los nombres de los objetos de destino para los que guarda una copia del informe de evaluación.

El siguiente ejemplo de código guarda una copia del informe de evaluación en el archivo `c:\sct\ar.pdf`.

```
SaveReportPDF
 -file:'c:\sct\ar.pdf'
/
```

El siguiente ejemplo de código guarda una copia del informe de evaluación en archivos CVS en la carpeta `c:\sct`.

```
SaveReportCSV
 -file:'c:\sct'
/
```

Para obtener más información sobre los comandos `SaveReportPDF` y `SaveReportCSV`, consulte la Referencia de la CLI de AWS Schema Conversion Tool.

Paso 4: Migrar el clúster de Apache Hadoop a Amazon EMR con AWS SCT

Tras configurar el proyecto de AWS SCT, inicie la migración del clúster de Apache Hadoop en las instalaciones a la Nube de AWS.

En este paso, utilice los comandos `Migrate`, `MigrationStatus` y `ResumeMigration`.

El comando `Migrate` migra los objetos de origen al clúster de destino. Este comando utiliza cuatro parámetros. Especifique el parámetro `filter` o `treePath`. El resto de los parámetros son opcionales.

- `filter`: el nombre del filtro que creó anteriormente para definir el alcance de los objetos de origen que desee migrar.
- `treePath`: la ruta explícita a los objetos de la base de datos de origen para los que guarda una copia del informe de evaluación.

- `forceLoad`: cuando se establece en `true`, AWS SCT carga automáticamente los árboles de metadatos de la base de datos durante la migración. El valor predeterminado es `false`.
- `forceMigrate`: si se establece en `true`, AWS SCT continúa la migración aunque el proyecto incluya una carpeta de HDFS y una tabla de Hive que hagan referencia al mismo objeto. El valor predeterminado es `false`.

El comando `MigrationStatus` devuelve información sobre el progreso de la migración. Para ejecutar este comando, introduzca el nombre del proyecto de migración para el parámetro `name`. Especificó este nombre en el comando `CreateProject`.

El comando `ResumeMigration` reanuda la migración interrumpida que inició con el comando `Migrate`. El comando `ResumeMigration` no utiliza parámetros. Para reanudar la migración, debe conectarse a los clústeres de origen y destino. Para obtener más información, consulte Administración del proyecto de migración.

El siguiente ejemplo de código migra los datos del servicio de HDFS de origen a Amazon EMR.

```
Migrate
 -treePath: 'Clusters.HADOOP_SOURCE.HDFS_SOURCE'
 -forceMigrate: 'true'
/
```

Ejecución del script de la CLI

Cuando termine de editar el script de la CLI de AWS SCT, guárdelo como un archivo con la extensión `.scts`. Ahora, puede ejecutar el script desde la carpeta `app` de la ruta de instalación de AWS SCT. Para ello, utilice el siguiente comando.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\hadoop.scts"
```

En el ejemplo anterior, sustituya *`script_path`* por la ruta del archivo con el script de la CLI. Para obtener más información sobre la ejecución de scripts de CLI en AWS SCT, consulte Modo script.

Administrar el proyecto de migración de macrodatos

Tras completar la migración, puede guardar y editar el proyecto de AWS SCT para usarlo en el futuro.

Para guardar el proyecto de AWS SCT, utilice el comando `SaveProject`. El comando no utiliza parámetros.

En el siguiente ejemplo de código se guarda el proyecto de AWS SCT.

```
SaveProject
/
```

Para abrir el proyecto de AWS SCT, utilice el comando `OpenProject`. Este comando usa un parámetro obligatorio. Para el parámetro `file`, introduzca el nombre y la ruta al archivo del proyecto de AWS SCT. Especificó el nombre del proyecto en el comando `CreateProject`. Agregue la extensión `.scts` al nombre del archivo de proyecto para ejecutar el comando `OpenProject`.

El siguiente ejemplo de código abre el proyecto de `hadoop_emr` desde la carpeta `c:\sct`.

```
OpenProject
 -file: 'c:\sct\hadoop_emr.scts'
/
```

Tras abrir el proyecto de AWS SCT, no es necesario agregar los clústeres de origen y destino porque ya los ha añadido al proyecto. Para empezar a trabajar con los clústeres de origen y destino, debe conectarse a ellos. Para ello, utilice los comandos `ConnectSourceCluster` y `ConnectTargetCluster`. Estos comandos utilizan los mismos parámetros que los comandos `AddSourceCluster` y `AddTargetCluster`. Puede editar el script de la CLI y reemplazar el nombre de estos comandos dejando la lista de parámetros sin cambios.

El siguiente ejemplo de código se conecta al clúster de Hadoop de origen.

```
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'
/
```

El siguiente ejemplo de código se conecta al clúster de Amazon EMR de destino.

```
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'
/
```

En el ejemplo anterior, sustituya *hadoop_address* por la dirección IP de su clúster de Hadoop. Si es necesario, configure el valor de la variable de puerto. A continuación, sustituya *hadoop_user* y *hadoop_password* por el nombre de usuario de Hadoop y la contraseña de este usuario. En *ruta\name*, introduzca el nombre y la ruta del archivo PEM del clúster de Hadoop de origen. Para obtener más información acerca de cómo agregar los clústeres de origen y destino, consulte Uso de Apache Hadoop como origen para AWS SCT.

Tras conectarse a los clústeres de Hadoop de origen y destino, debe conectarse a los servicios de Hive y HDFS, así como al bucket de Amazon S3. Para ello, utilice los comandos `ConnectSourceClusterHive`, `ConnectSourceClusterHdfs`, `ConnectTargetClusterHive`, `ConnectTargetClusterHdfs` y `ConnectTargetClusterS3`. Estos comandos utilizan los mismos parámetros que los comandos que utilizó para agregar los servicios de Hive y HDFS y el bucket de Amazon S3 al proyecto. Edite el script de la CLI para reemplazar el prefijo `Add` por `Connect` en los nombres de los comandos.

Conversión de Apache Oozie a AWS Step Functions con AWS Schema Conversion Tool

Para convertir los flujos de trabajo de Apache Oozie, utilice la versión 1.0.671 o posterior de AWS SCT. Además, familiarícese con la interfaz de la línea de comandos (CLI) de AWS SCT. Para obtener más información, consulte AWS SCT Referencia CLI.

Temas

- Descripción general de la conversión
- Paso 1: Conectar con los servicios de origen y destino
- Paso 2: Configurar las reglas de asignación
- Paso 3: Configurar los parámetros
- Paso 4: Crear un informe de evaluación
- Paso 5: Convertir los flujos de trabajo de Apache Oozie a AWS Step Functions con AWS SCT
- Ejecución del script de la CLI
- Nodos de Apache Oozie que AWS SCT puede convertir a AWS Step Functions

Descripción general de la conversión

El código fuente de Apache Oozie incluye nodos de acción, nodos de flujo de control y propiedades de trabajo. Los nodos de acción definen los trabajos que se ejecutan en el flujo de trabajo de Apache Oozie. Cuando utiliza Apache Oozie para organizar el clúster de Apache Hadoop, un nodo de acción incluye un trabajo de Hadoop. Los nodos de flujo de control proporcionan un mecanismo para controlar la ruta del flujo de trabajo. Los nodos de flujo de control incluyen nodos tales como `start`, `end`, `decision`, `fork` y `join`.

AWS SCT convierte los nodos de acción de origen y los nodos de flujo de control a AWS Step Functions. En AWS Step Functions, define los flujos de trabajo en Amazon States Language (ASL). AWS SCT usa ASL para definir la máquina de estado, que es un conjunto de estados que realizan tareas, determinan a qué estados se debe pasar a continuación, detener cuando se produce un error, etcétera. A continuación, AWS SCT carga los archivos JSON con las definiciones de las máquinas de estado. Luego, AWS SCT puede usar el rol AWS Identity and Access Management (de IAM) para configurar las máquinas de estado en AWS Step Functions. Para obtener más información, consulte ¿Qué es AWS Step Functions? en la Guía para desarrolladores de AWS Step Functions.

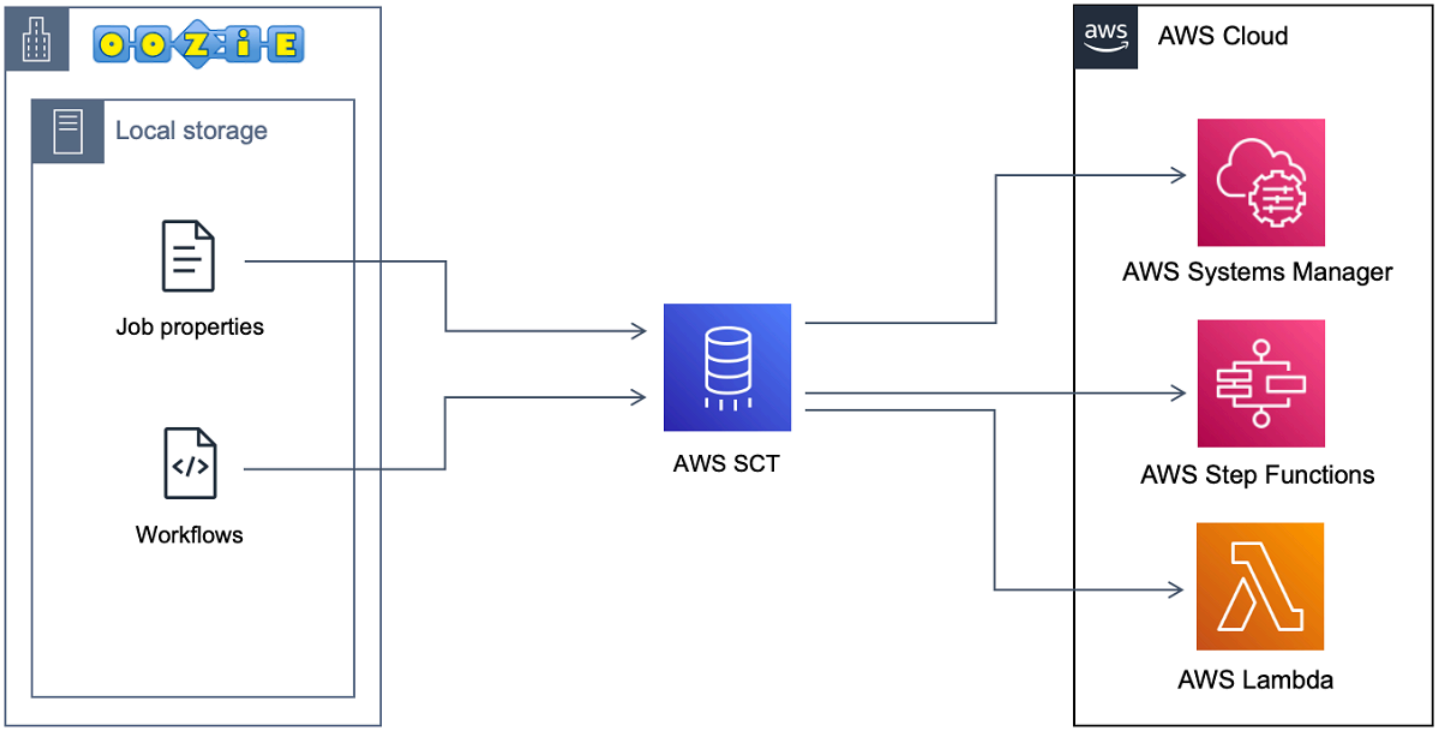
Además, AWS SCT crea un paquete de extensiones con funciones de AWS Lambda que emulan las funciones de origen que AWS Step Functions no admite. Para obtener más información, consulte Uso de paquetes AWS SCT de extensión.

AWS SCT migra las propiedades del trabajo de origen a AWS Systems Manager. Para almacenar los nombres y valores de los parámetros, AWS SCT utiliza Parameter Store, una capacidad de AWS

Systems Manager. Para obtener más información, consulte ¿Qué es AWS Systems Manager? en la Guía del usuario de AWS Systems Manager.

Puede utilizar AWS SCT para actualizar automáticamente los valores y los nombres de los parámetros. Debido a las diferencias de arquitectura entre Apache Oozie y AWS Step Functions, es posible que tenga que configurar los parámetros. AWS SCTpuede encontrar un nombre o valor de parámetro específico en los archivos de origen y sustituirlos por valores nuevos. Para obtener más información, consulte Paso 3: Configurar los parámetros.

En la siguiente imagen se muestra el diagrama de la arquitectura de conversión de Apache Oozie a AWS Step Functions.



Para iniciar la conversión, cree y ejecute el script de la CLI de AWS SCT. Este script incluye el conjunto completo de comandos para ejecutar la conversión. Puede descargar y editar una plantilla del script de conversión de Apache Oozie. Para obtener más información, consulte Obtención de escenarios de la CLI.

Asegúrese de que el script incluya los pasos siguientes.

Paso 1: Conectar con los servicios de origen y destino

Para iniciar la conversión de su clúster de Apache Oozie, cree un proyecto de AWS SCT nuevo. A continuación, conecte con los servicios de origen y destino. Cree y aprovisione los recursos de AWS

de destino antes de iniciar la migración. Para obtener más información, consulte Requisitos previos para utilizar Apache Oozie como origen.

En este paso, utilice los siguientes comandos de la CLI de AWS SCT.

- `CreateProject`: para crear un proyecto de AWS SCT nuevo.
- `AddSource`: para agregar los archivos de Apache Oozie de origen al proyecto de AWS SCT.
- `ConnectSource`: para conectar con Apache Oozie como origen.
- `AddTarget`: para agregar AWS Step Functions como destino de migración al proyecto.
- `ConnectTarget`: para conectar con AWS Step Functions.

Para obtener ejemplos de uso de estos comandos de la CLI de AWS SCT, consulte Usar Apache Oozie como origen.

Cuando ejecuta los comandos `ConnectSource` o `ConnectTarget`, AWS SCT intenta establecer la conexión con sus servicios. Si el intento de conexión falla, AWS SCT deja de ejecutar los comandos del script de la CLI y muestra un mensaje de error.

Paso 2: Configurar las reglas de asignación

Después de conectarse a los servicios de origen y destino, configure las reglas de asignación. Una regla de asignación define el destino de migración para los flujos de trabajo y parámetros de Apache Oozie de origen. Para obtener más información acerca de las reglas de asignación, consulte Creación de reglas de asignación en AWS SCT.

Para definir los objetos de origen y destino para la conversión, utilice el comando `AddServerMapping`. Este comando utiliza dos parámetros: `sourceTreePath` y `targetTreePath`. Los valores de estos parámetros incluyen una ruta explícita a los objetos de origen y destino. Para la conversión de Apache Oozie a AWS Step Functions, estos parámetros deben empezar por `ETL`.

El siguiente ejemplo de código crea una regla de asignación para los objetos `OOZIE` y `AWS_STEP_FUNCTIONS`. En el paso anterior, agregó estos objetos al proyecto de AWS SCTmediante los comandos `AddSource` y `AddTarget`.

```
AddServerMapping
    -sourceTreePath: 'ETL.APACHE_OOZIE'
    -targetTreePath: 'ETL.AWS_STEP_FUNCTIONS'
```

```
/
```

Para obtener más información sobre el comando `AddServerMapping`, consulte la Referencia de la CLI de AWS Schema Conversion Tool.

Paso 3: Configurar los parámetros

Si los flujos de trabajo de Apache Oozie de origen utilizan parámetros, es posible que tenga que cambiar sus valores después de la conversión a AWS Step Functions. Además, es posible que tenga que agregar parámetros nuevos para usar con su AWS Step Functions.

En este paso, utilice los comandos `AddParameterMapping` y `AddTargetParameter`.

Para reemplazar los valores de los parámetros en los archivos de origen, utilice el comando `AddParameterMapping`. AWS SCT analiza los archivos de origen, busca los parámetros por nombre o valor y cambia los valores. Puede ejecutar un solo comando para analizar todos los archivos de origen. Para definir el alcance de los archivos que se van a analizar, utilice uno de los tres primeros parámetros de la siguiente lista. Este comando utiliza hasta seis parámetros.

- `filterName`: el nombre del filtro de los objetos de origen. Puede crear un filtro mediante el comando `CreateFilter`.
- `treePath`: la ruta explícita a los objetos de origen.
- `namePath`: la ruta explícita a un objeto de origen concreto.
- `sourceParameterName`: el nombre del parámetro de origen.
- `sourceValue`: el valor del parámetro de origen.
- `targetValue`: el valor del parámetro de destino.

El siguiente ejemplo de código reemplaza todos los parámetros en los que el valor es igual a `c:\oozie\hive.py` por el valor `s3://bucket-oozie/hive.py`.

```
AddParameterMapping
 -treePath: 'ETL.OOZIE.Applications'
 -sourceValue: 'c:\oozie\hive.py'
 -targetValue: 's3://bucket-oozie/hive.py'
/
```

El siguiente ejemplo de código reemplaza todos los parámetros en los que el nombre es igual a nameNode por el valor `hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020`.

```
AddParameterMapping
    -treePath: 'ETL.OOZIE_SOURCE.Applications'
    -sourceParameter: 'nameNode'
    -targetValue: 'hdfs://ip-111-222-33-44.eu-west-1.compute.internal:8020'
/
```

El siguiente ejemplo de código reemplaza todos los parámetros en los que el nombre es igual a `nameNode` y el valor es igual a `hdfs://ip-55.eu-west-1.compute.internal:8020` por el valor del parámetro `targetValue`.

```
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'
/
```

Para agregar un parámetro nuevo a los archivos de destino además de un parámetro existente en los archivos de origen, utilice el comando `AddTargetParameter`. Este comando usa el mismo conjunto de parámetros que el comando `AddParameterMapping`.

El siguiente ejemplo de código agrega el parámetro de destino `clusterId` en lugar del parámetro `nameNode`.

```
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'
/
```

Para obtener más información sobre los comandos `AddServerMapping`, `AddParameterMapping`, `AddTargetParameter` y `CreateFilter`, consulte la Referencia de la CLI de AWS Schema Conversion Tool.

Paso 4: Crear un informe de evaluación

Antes de iniciar la conversión, le recomendamos que cree un informe de evaluación. Este informe resume todas las tareas de migración y detalla las acciones que surgirán durante la migración. Para

asegurarse de que la migración no falle, consulte este informe y aborde las medidas que se deben tomar antes de la migración. Para obtener más información, consulte Informes de evaluación de migración.

En este paso, utilice el comando `CreateReport`. Este comando emplea dos parámetros. El primer parámetro describe los objetos de origen para los que AWS SCT crea un informe de evaluación. Para ello, utilice uno de los siguientes parámetros: `filterName`, `treePath` o `namePath`. Este parámetro es obligatorio. También puede agregar un parámetro booleano opcional `forceLoad`. Si establece este parámetro en `true`, AWS SCT carga automáticamente todos los objetos secundarios del objeto de origen que especifique en el comando `CreateReport`.

El siguiente ejemplo de código crea un informe de evaluación para el nodo `Applications` de los archivos Oozie de origen.

```
CreateReport
    -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

Puede guardar una copia del informe de evaluación en formato PDF o como un archivo de valores separados por comas (CSV). Para ello, utilice el comando `SaveReportPDF` o `SaveReportCSV`.

El comando `SaveReportPDF` guarda una copia del informe de evaluación en un archivo PDF. Este comando utiliza cuatro parámetros. El parámetro `file` es obligatorio, mientras que otros parámetros son opcionales.

- `file`: la ruta al archivo PDF y su nombre.
- `filter`: el nombre del filtro que creó anteriormente para definir el alcance de los objetos de origen que desee migrar.
- `treePath`: la ruta explícita a los objetos de la base de datos de origen para los que guarda una copia del informe de evaluación.
- `namePath`: la ruta que incluye solo los nombres de los objetos de destino para los que guarda una copia del informe de evaluación.

El comando `SaveReportCSV` guarda el informe de evaluación en archivos CSV. Este comando utiliza cuatro parámetros. El parámetro `directory` es obligatorio, mientras que otros parámetros son opcionales.

- `directory`: la ruta a la carpeta en la que AWS SCT guarda los archivos CSV.

- `filter`: el nombre del filtro que creó anteriormente para definir el alcance de los objetos de origen que desee migrar.
- `treePath`: la ruta explícita a los objetos de la base de datos de origen para los que guarda una copia del informe de evaluación.
- `namePath`: la ruta que incluye solo los nombres de los objetos de destino para los que guarda una copia del informe de evaluación.

El siguiente ejemplo de código guarda una copia del informe de evaluación en el archivo `c:\sct\ar.pdf`.

```
SaveReportPDF
 -file:'c:\sct\ar.pdf'
/
```

El siguiente ejemplo de código guarda una copia del informe de evaluación en archivos CVS en la carpeta `c:\sct`.

```
SaveReportCSV
 -file:'c:\sct'
/
```

Para obtener más información sobre los comandos `CreateReport`, `SaveReportPDF` y `SaveReportCSV`, consulte la Referencia de la CLI de AWS Schema Conversion Tool.

Paso 5: Convertir los flujos de trabajo de Apache Oozie a AWS Step Functions con AWS SCT

Después de configurar su proyecto de AWS SCT, convierta el código fuente y aplíquelo a la Nube de AWS.

En este paso, utilice los comandos `Convert`, `SaveOnS3`, `ConfigureStateMachine` y `ApplyToTarget`.

El comando `Migrate` migra los objetos de origen al clúster de destino. Este comando utiliza cuatro parámetros. Especifique el parámetro `filter` o `treePath`. El resto de los parámetros son opcionales.

- `filter`: el nombre del filtro que creó anteriormente para definir el alcance de los objetos de origen que desee migrar.
- `namePath`: la ruta explícita a un objeto de origen concreto.
- `treePath`: la ruta explícita a los objetos de la base de datos de origen para los que guarda una copia del informe de evaluación.
- `forceLoad`: cuando se establece en `true`, AWS SCT carga automáticamente los árboles de metadatos de la base de datos durante la migración. El valor predeterminado es `false`.

El siguiente ejemplo de código convierte los archivos de la carpeta `Applications` a archivos Oozie de origen.

```
Convert
    -treePath: 'ETL.APACHE_OOZIE.Applications'
/
```

`SaveOnS3` carga las definiciones de las máquinas de estado en el bucket de Amazon S3. Este comando utiliza el parámetro `treePath`. Para ejecutar este comando, utilice la carpeta de destino con las definiciones de las máquinas de estado como valor de este parámetro.

El siguiente ejemplo carga la carpeta `State machine definitions` del objeto de destino de `AWS_STEP_FUNCTIONS` en el bucket de Amazon S3. AWS SCT utiliza el bucket de Amazon S3 que guardó en el perfil de servicio de AWS en el paso Requisitos previos.

```
SaveOnS3
    -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
/
```

El comando `ConfigureStateMachine` configura las máquinas de estado. Este comando utiliza hasta seis parámetros. Para definir el alcance de los archivos que se van a analizar, utilice uno de los tres primeros parámetros de la siguiente lista.

- `filterName`: el nombre del filtro de los objetos de destino. Puede crear un filtro mediante el comando `CreateFilter`.
- `treePath`: la ruta explícita a los objetos de destino.
- `namePath`: la ruta explícita a un objeto de destino concreto.
- `iamRole`: el nombre de recurso de Amazon (ARN) del rol de IAM que proporciona acceso a las máquinas de estado. Este parámetro es obligatorio.

El siguiente ejemplo de código configura las máquinas de estado definidas en AWS_STEP_FUNCTIONS utilizando el rol de IAM *role_name*.

```
ConfigureStateMachine
    -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machine definitions'
    -role: 'arn:aws:iam::555555555555:role/role_name'
/
```

El comando ApplyToTarget aplica el código convertido al servidor de destino. Para ejecutar este comando, utilice uno de los siguientes parámetros: filterName, treePath o namePath para definir los objetos de destino que se van a aplicar.

El siguiente ejemplo de código aplica la máquina de estado app_wp a AWS Step Functions.

```
ApplyToTarget
    -treePath: 'ETL.AWS_STEP_FUNCTIONS.State machines.app_wp'
/
```

Utilice un paquete de extensión de AWS SCT para asegurarse de que el código convertido produce los mismos resultados que el código de origen. Se trata de un conjunto de funciones de AWS Lambda que emulan las funciones de Apache Oozie que AWS Step Functions no admite. Para instalar este paquete de extensión, puede usar el comando CreateLambdaExtPack.

Este comando utiliza hasta cinco parámetros. Utilice **Oozie2SF** para extPackId. En este caso, AWS SCT crea un paquete de extensión para las funciones de Apache Oozie de origen.

- extPackId: el identificador único de un conjunto de funciones de Lambda. Este parámetro es obligatorio.
- tempDirectory: la ruta donde en la que AWS SCT puede almacenar los archivos temporales. Este parámetro es obligatorio.
- awsProfile: el nombre de su perfil de AWS.
- lambdaExecRoles: la lista de nombres de recursos de Amazon (ARN) de los roles de ejecución que se van a utilizar para las funciones de Lambda.
- createInvokeRoleFlag: el indicador booleano que indica si se debe crear un rol de ejecución para. AWS Step Functions

Para instalar y usar el paquete de extensión, proporcione los permisos necesarios. Para obtener más información, consulte Permisos para usar funciones de AWS Lambda en el paquete de extensión.

Para obtener más información sobre los comandos `Convert`, `SaveOnS3`, `ConfigureStateMachine`, `ApplyToTarget` y `CreateLambdaExtPack`, consulte la Referencia de la CLI de AWS Schema Conversion Tool.

Ejecución del script de la CLI

Cuando termine de editar el script de la CLI de AWS SCT, guárdelo como un archivo con la extensión `.scts`. Ahora, puede ejecutar el script desde la carpeta app de la ruta de instalación de AWS SCT. Para ello, utilice el siguiente comando.

```
RunSCTBatch.cmd --pathtoscts "C:\script_path\oozie.scts"
```

En el ejemplo anterior, sustituya *`script_path`* por la ruta del archivo con el script de la CLI. Para obtener más información sobre la ejecución de scripts de CLI en AWS SCT, consulte Modo script.

Nodos de Apache Oozie que AWS SCT puede convertir a AWS Step Functions

Puede usar AWS SCT para convertir nodos de acción y de flujo de control de Apache Oozie a AWS Step Functions.

Entre los nodos de acción compatibles se incluyen:

- Acción de Hive
- Acción de Hive2
- Acción de Spark
- Acción de MapReduce Streaming
- Acción de Java
- Acción de DistCP
- Acción de Piga
- Acción de Sqoop
- Acción de FS
- Acción de Shell

Entre los nodos de control de flujo compatibles se incluyen:

- Acción Start
- Acción End
- Acción Kit
- Acción Decision
- Acción Fork
- Acción Join

Uso de AWS SCT con AWS DMS

Uso de un agente de replicación de AWS SCT con AWS DMS

Para migraciones de bases de datos muy grandes, puede utilizar un agente de replicación de AWS SCT (aws-schema-conversion-tool-dms-agent) para copiar datos de la base de datos en las instalaciones a Amazon S3 o un dispositivo AWS Snowball Edge. El agente de replicación funciona con AWS DMS y puede funcionar en segundo plano mientras AWS SCT está cerrado.

Cuando se trabaja con AWS Snowball Edge, el agente de AWS SCT replica los datos en el dispositivo AWS Snowball. El dispositivo se envía a AWS y los datos se cargan en un bucket de Amazon S3. Durante este tiempo, el agente de AWS SCT sigue en ejecución. A continuación, el agente obtiene los datos de Amazon S3 y los copia en el punto de enlace de destino.

Para obtener más información, consulte Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift.

Uso de un agente de extracción de datos de AWS SCT con AWS DMS

En AWS SCT, encontrará un agente de extracción de datos (`aws-schema-conversion-tool-extractor`) que le ayudará a facilitar las migraciones de Apache Cassandra a Amazon DynamoDB. Cassandra y DynamoDB son bases de datos NoSQL, pero difieren en la arquitectura del sistema y en la representación de los datos. Puede utilizar flujos de trabajo basados en asistentes de AWS SCT para automatizar el proceso de migración de Cassandra a DynamoDB. AWS SCT se integra con AWS Database Migration Service (AWS DMS) para realizar la migración real.

Para obtener más información, consulte Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift.

Cómo aumentar los niveles de registro cuando se usa AWS SCT con AWS DMS

Puede aumentar los niveles de registro cuando utilice AWS SCT conAWS DMS, por ejemplo, si necesita trabajar con AWS Support.

Después de instalar AWS SCT y los controladores necesarios, abra la aplicación seleccionando el icono AWS SCT. Si ve una notificación de actualización, puede optar por actualizarla antes o después de que se complete el proyecto. Si se abre una ventana de proyecto automático, ciérrela y cree un proyecto manualmente.

Para aumentar los niveles de registro cuando utilice AWS SCT con AWS DMS

1. En el menú Configuración, elija Configuración global.
2. En la ventana Configuración global, seleccione Registro.
3. En Modo de depuración, seleccione True.
4. En la sección Nivel de mensaje, puede modificar los siguientes tipos de registros:

   - General
   - Programa de carga
   - Analizador
   - Impresora
   - Solucionador
   - Telemetría
   - Conversor

   De forma predeterminada, todos los niveles de mensajes están configurados en Información.
5. Elija un nivel de registro para cualquier tipo de nivel de mensaje que desee modificar:

   - Rastreo (registro más detallado)
   - Debug
   - Información
   - Advertencia
   - Error (registro menos detallado)
   - Critico
   - Obligatorio
6. Seleccione Aplicar para modificar la configuración del proyecto.
7. Seleccione Aceptar para cerrar el cuadro de diálogo Configuración global.

Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift

Puede usar un AWS SCT agente para extraer datos de su almacén de datos local y migrarlos a Amazon Redshift. El agente extrae los datos y los carga en Amazon S3 o, en el caso de migraciones a gran escala, en un dispositivo AWS Snowball Edge. A continuación, puede utilizar un AWS SCT agente para copiar los datos a Amazon Redshift.

Como alternativa, puede usar AWS Database Migration Service (AWS DMS) para migrar datos a Amazon Redshift. La ventaja de AWS DMS es que admite replicación continua (captura de datos de cambio). Sin embargo, para aumentar la velocidad de migración de datos, utilice varios AWS SCT agentes en paralelo. Según nuestras pruebas, los AWS SCT agentes migran los datos más rápido que entre un 15 y un 35 AWS DMS por ciento. La diferencia de velocidad se debe a la compresión de datos, al soporte de migración de particiones de tabla en paralelo y a los diferentes ajustes de configuración. Para obtener más información, consulte Uso de una base de datos de Amazon Redshift como destino de AWS Database Migration Service.

Amazon S3 es un servicio de almacenamiento y de recuperación. Para almacenar un objeto en Amazon S3, debe cargar en un bucket de Amazon S3 el archivo que quiera almacenar. Al cargar un archivo, puede configurar permisos en el objeto y también en cualquier metadato.

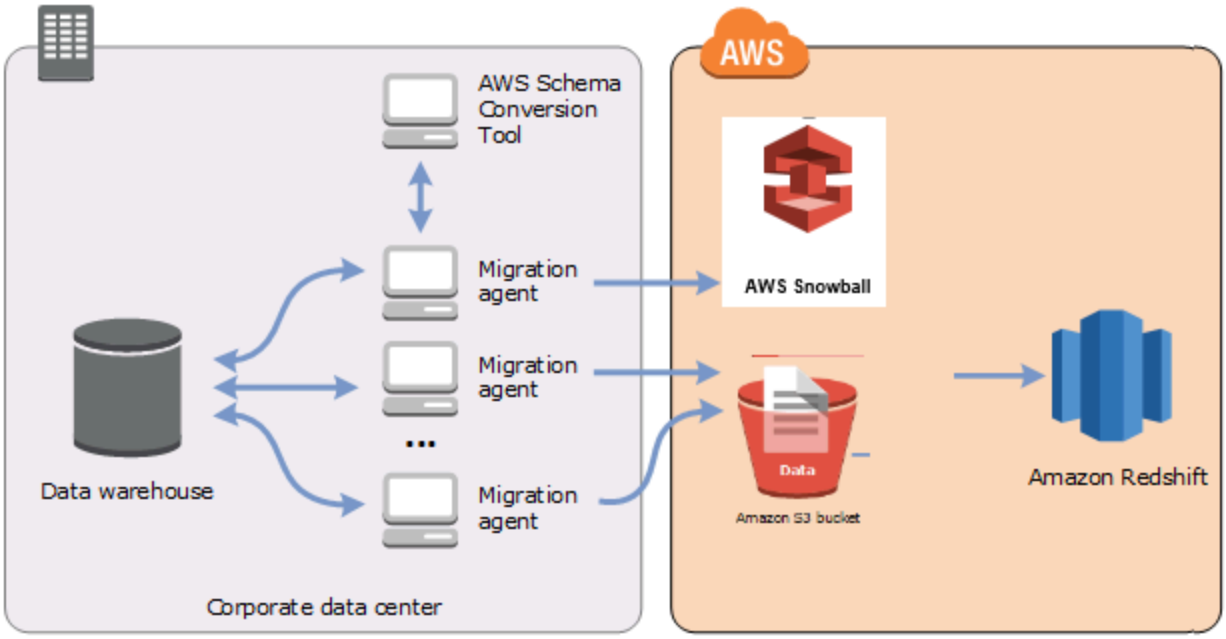
Migraciones a gran escala

Las migraciones de datos a gran escala pueden incluir muchos terabytes de información y pueden ralentizarse debido al rendimiento de la red y a la enorme cantidad de datos que hay que mover. AWS Snowball Edge es un AWS servicio que puede utilizar para transferir datos a la nube a gran faster-than-network velocidad mediante un AWS dispositivo propio. Un dispositivo AWS Snowball Edge puede almacenar hasta 100 TB de datos. Utiliza un cifrado de 256 bits y un módulo de plataforma segura (TPM) estándar del sector para garantizar la seguridad y la integridad de sus datos. chain-of-custody AWS SCT funciona con dispositivos Edge. AWS Snowball

Cuando se utiliza AWS SCT un dispositivo AWS Snowball Edge, se migran los datos en dos etapas. En primer lugar, se procesan AWS SCT los datos de forma local y, a continuación, se mueven esos datos al dispositivo AWS Snowball Edge. A continuación, envía el dispositivo a AWS mediante el proceso AWS Snowball Edge y, a continuación, carga AWS automáticamente los datos en un bucket de Amazon S3. A continuación, cuando los datos estén disponibles en Amazon S3, se AWS SCT

suelen migrar a Amazon Redshift. Los agentes de extracción de datos pueden funcionar en segundo plano mientras AWS SCT está cerrado.

En el siguiente diagrama se muestran los escenarios admitidos.



En la actualidad, los agentes de extracción de datos se admiten para los siguientes data warehouses de origen:

- Azure Synapse Analytics
- BigQuery
- Greenplum Database (versión 4.3)
- Microsoft SQL Server (versión 2008 y posteriores)
- Netezza (versión 7.0.3 y posteriores)
- Oracle (versión 10 y posteriores)
- Snowflake (versión 3)
- Teradata (versión 13 y posteriores)
- Vertica (versión 7.2.2 y posteriores)

Puede conectarse a puntos de conexión de FIPS para Amazon Redshift si tiene que cumplir con los requisitos de seguridad del Estándar federal de procesamiento de la información (FIPS). Los puntos finales de FIPS están disponibles en las siguientes regiones: AWS

- Región Este de EE. UU. (Norte de Virginia) (redshift-fips.us-east-1.amazonaws.com)
- Región Este de EE. UU. (Ohio) (redshift-fips.us-east-2.amazonaws.com)
- Región Oeste de EE. UU. (Norte de California) (redshift-fips.us-west-1.amazonaws.com)
- Región Oeste de EE. UU. (Oregón) (redshift-fips.us-west-2.amazonaws.com)

Utilice la información en los temas siguientes para aprender a trabajar con los agentes de extracción de datos.

Temas

- Requisitos previos para utilizar agentes de extracción de datos
- Instalación de agentes de extracción
- Configuración de agentes de extracción
- Registrar los agentes de extracción con el AWS Schema Conversion Tool
- Ocultar y recuperar la información de un AWS SCT agente
- Crear reglas de migración de datos en AWS SCT
- Cambio de los ajustes de extracción y copia en la configuración del proyecto
- Ordenar los datos antes de migrarlos mediante AWS SCT
- Crear, ejecutar y supervisar una tarea de extracción de AWS SCT datos
- Exportación e importación de una tarea AWS SCT de extracción de datos
- Extracción de datos mediante un dispositivo AWS Snowball Edge
- Resultado de la tarea de extracción de datos
- Uso de particiones virtuales con AWS Schema Conversion Tool
- Uso de particiones nativas
- Migración de los LOB a Amazon Redshift
- Prácticas recomendadas y solución de problemas de agentes de extracción de datos

Requisitos previos para utilizar agentes de extracción de datos

Antes de trabajar con agentes de extracción de datos, añada los permisos necesarios para Amazon Redshift como destino al usuario de Amazon Redshift. Para obtener más información, consulte Permisos para Amazon Redshift como destino.

A continuación, almacene la información del bucket de Amazon S3 y configure el almacén de confianza y de claves de la capa de sockets seguros (SSL).

Configuración de Amazon S3

Después de que sus agentes extraigan los datos, los cargarán en el bucket de Amazon S3. Antes de continuar, debe proporcionar las credenciales para conectarse a su AWS cuenta y a su bucket de Amazon S3. Almacena sus credenciales y la información del bucket en un perfil en la configuración global de la aplicación y, a continuación, asocia el perfil a su AWS SCT proyecto. Si fuera necesario, seleccione Configuración global para crear un nuevo perfil. Para obtener más información, consulte Almacenamiento de perfiles de servicio de AWS en AWS SCT.

Para migrar datos a la base de datos de Amazon Redshift de destino, el agente de extracción de AWS SCT datos necesita permiso para acceder al bucket de Amazon S3 en su nombre. Para conceder este permiso, cree un usuario AWS Identity and Access Management (IAM) con la siguiente política.

```
{
    "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"
        }
    ]
}
```

En el ejemplo anterior, reemplace *bucket_name* por el nombre del bucket de Amazon S3. A continuación, sustituya *111122223333:user/DataExtractionAgentName* por el nombre del usuario de IAM.

Asunción de roles de IAM

Para mayor seguridad, puede utilizar funciones AWS Identity and Access Management (IAM) para acceder a su bucket de Amazon S3. Para ello, cree un usuario de IAM para sus agentes de extracción de datos sin ningún permiso. A continuación, cree un rol de IAM que permita el acceso a Amazon S3 y especifique la lista de servicios y usuarios que pueden asumir este rol. Para obtener más información, consulte Roles de IAM en la Guía del usuario de IAM.

Para configurar roles de IAM para acceder al bucket de S3

1. Cree un usuario de IAM nuevo. En Credenciales de usuario, seleccione el tipo Acceso mediante programación.

2. Configure el entorno anfitrión para que su agente de extracción de datos pueda asumir la función que AWS SCT le proporciona. Asegúrese de que el usuario que configuró en el paso anterior permita que los agentes de extracción de datos utilicen la cadena de proveedores de credenciales. Para obtener más información, consulte Uso de credenciales en la Guía para desarrolladores de AWS SDK for Java .
3. Cree un nuevo rol de IAM que tenga acceso al bucket de Amazon S3.
4. Modifique la sección de confianza de este rol para confiar en el usuario que creó antes de asumir el rol. En el ejemplo siguiente, sustituya *111122223333:user/DataExtractionAgentName* por el nombre de su usuario.

```
{
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::111122223333:user/DataExtractionAgentName"
    },
    "Action": "sts:AssumeRole"
}
```

5. Modifique la sección de confianza de este rol para confiar en `redshift.amazonaws.com` para asumir el rol.

```
{
    "Effect": "Allow",
    "Principal": {
        "Service": [
            "redshift.amazonaws.com"
        ]
    },
    "Action": "sts:AssumeRole"
}
```

6. Adjunte este rol al clúster de Amazon Redshift.

Ahora puede ejecutar el agente de extracción de datos en AWS SCT.

Cuando se utiliza la función de asumir un rol de IAM, la migración de datos funciona de la siguiente manera. El agente de extracción de datos se inicia y obtiene las credenciales de usuario mediante la cadena de proveedores de credenciales. A continuación, cree una tarea de migración de datos en AWS SCT, a continuación, especifique la función de IAM que deben asumir los agentes de

extracción de datos e inicie la tarea. AWS Security Token Service (AWS STS) genera credenciales temporales para acceder a Amazon S3. El agente de extracción de datos utiliza estas credenciales para cargar datos en Amazon S3.

A continuación, AWS SCT proporciona a Amazon Redshift la función de IAM. A su vez, Amazon Redshift obtiene nuevas credenciales temporales AWS STS para acceder a Amazon S3. Amazon Redshift utiliza estas credenciales para copiar datos desde Amazon S3 a la tabla de Amazon Redshift.

Configuración de seguridad

Los agentes de extracción AWS Schema Conversion Tool y los agentes de extracción pueden comunicarse a través de Secure Sockets Layer (SSL). Para habilitar la SSL, configure un almacén de confianza y un almacén de claves.

Para configurar una comunicación segura con el agente de extracción

1. Inicie el AWS Schema Conversion Tool.
2. Abra el menú Configuración y seleccione Configuración global. Aparecerá el cuadro de diálogo Configuración global.
3. Elija Seguridad.
4. Seleccione Generar almacén de confianza y claves o haga clic en Seleccionar almacén de confianza y claves existente.

   Si selecciona Generar almacén de confianza y de claves, a continuación tendrá que especificar el nombre y la contraseña de los almacenes de confianza y de claves y la ruta a la ubicación de los archivos generados. Utilizará estos archivos en pasos posteriores.

   Si selecciona Seleccionar almacén de confianza y claves existente, después especificará la contraseña y el nombre del archivo para los almacenes de confianza y de claves. Utilizará estos archivos en pasos posteriores.
5. Tras haber especificado el almacén de confianza y el almacén de claves, seleccione Aceptar para cerrar el cuadro de diálogo Configuración global.

Configuración del entorno para los agentes de extracción de datos

Puede instalar varios agentes de extracción de datos en un único host. Sin embargo, recomendamos que ejecute un agente de extracción de datos en un host.

Para ejecutar el agente de extracción de datos, asegúrese de utilizar un host con al menos cuatro vCPU y 32 GB de memoria. Además, establezca la memoria mínima disponible en AWS SCT al menos cuatro GB. Para obtener más información, consulte Configurar memoria adicional.

La configuración óptima y la cantidad de hosts de agentes dependen de la situación específica de cada cliente. Asegúrese de tener en cuenta factores como la cantidad de datos que se van a migrar, el ancho de banda de la red, el tiempo de extracción de los datos, etc. En primer lugar, puede realizar una prueba de concepto (PoC) y, a continuación, configurar los agentes de extracción de datos y los hosts de acuerdo con los resultados de esta PoC.

Instalación de agentes de extracción

Le recomendamos que instale varios agentes de extracción en equipos individuales, independientes del equipo en el que se esté ejecutando la AWS Schema Conversion Tool.

En la actualidad, los agentes de extracción son compatibles con los siguientes sistemas operativos:

- Microsoft Windows
- Red Hat Enterprise Linux (RHEL) 6.0
- Ubuntu Linux (versión 14.04 y posteriores)

Utilice el siguiente procedimiento para instalar agentes de extracción. Repita este procedimiento para cada equipo en el que desee instalar un agente de extracción.

Para instalar un agente de extracción

1. Si aún no ha descargado el archivo de AWS SCT instalación, siga las instrucciones que aparecen en Instalación, verificación y actualización AWS SCT para descargarlo. El archivo.zip que contiene el archivo de AWS SCT instalación también contiene el archivo de instalación del agente de extracción.
2. Descargue e instale la versión más reciente de Amazon Corretto 11. Para obtener más información, consulte Descargas para Amazon Corretto 11 en la Guía del Usuario de Amazon Corretto 11.
3. Localice el archivo del instalador para su agente de extracción en una subcarpeta llamada agentes. El archivo correcto para el sistema operativo del equipo en el que quiera instalar el agente de extracción se muestra a continuación.

Sistema operativo	Nombre de archivo
Microsoft Windows	`aws-schema-conversion-tool-extractor -2.0.1. build-number .msi`
RHEL	`aws-schema-conversion-tool-extractor -2.0.1. build-number .x86_64.rpm`
Ubuntu Linux	`aws-schema-conversion-tool-extractor -2.0.1. build-number .deb`

4. Instale el agente de extracción en un equipo independiente copiando el archivo del instalador en el nuevo equipo.
5. Ejecute el archivo del instalador. Utilice las instrucciones para su sistema operativo, que se muestran a continuación.

Sistema operativo	Instrucciones de instalación
Microsoft Windows	Haga doble clic en el archivo para ejecutar el instalador.

RHEL	Ejecute los siguientes comandos en la carpeta en la que haya descargado o movido el archivo.
	`sudo rpm -ivh aws-schema-conversion-tool-extractor -2.0.1. build-number .x86_64.rpm`
	`sudo ./sct-extractor-setup.sh --config`

Ubuntu Linux	Ejecute los siguientes comandos en la carpeta en la que haya descargado o movido el archivo.
	`sudo dpkg -i aws-schema-conversion-tool-extractor -2.0.1. build-number .deb`
	`sudo ./sct-extractor-setup.sh --config`

6. Seleccione Siguiente, acepte el acuerdo de licencia y elija Siguiente.
7. Introduzca la ruta para instalar el agente AWS SCT de extracción de datos y seleccione Siguiente.

8. Elija Instalar para instalar el agente de extracción de datos.

   AWS SCT instala el agente de extracción de datos. Para completar la instalación, configure el agente de extracción de datos. AWS SCT inicia automáticamente el programa de configuración. Para obtener más información, consulte Configuración de agentes de extracción.

9. Seleccione Finalizar para cerrar el asistente de instalación después de configurar el agente de extracción de datos.

Configuración de agentes de extracción

Utilice el siguiente procedimiento para configurar agentes de extracción. Repita este procedimiento para cada equipo en que tenga instalado un agente de extracción.

Para configurar su agente de extracción

1. Inicie el programa de configuración:

   - En Windows, AWS SCT inicia el programa de configuración automáticamente durante la instalación de un agente de extracción de datos.

     Si es necesario, puede iniciar el programa de configuración manualmente. Para ello, ejecute el archivo `ConfigAgent.bat` en Windows. Encontrará este archivo en la carpeta en la que instaló el agente.
   - En RHEL y Ubuntu, ejecute el archivo `sct-extractor-setup.sh` desde la ubicación en la que instaló el agente.

   El programa de instalación le pedirá información. Para cada solicitud, aparecerá un valor predeterminado.

2. Acepte el valor predeterminado en cada solicitud o introduzca un valor nuevo.

   Especifique la siguiente información:

   - En Puerto oyente, escriba el número de puerto en el que esté escuchando el agente.
   - En Agregar un proveedor de origen, introduzca sí y, a continuación, introduzca la plataforma de almacenamiento de datos de origen.
   - En Controlador JDBC, introduzca la ubicación en la que haya instalado los controladores JDBC.

- En la carpeta de trabajo, introduzca la ruta en la que el agente de extracción de AWS SCT datos almacenará los datos extraídos. La carpeta de trabajo puede estar en un equipo distinto al del agente, y una única carpeta de trabajo se puede compartir entre varios agentes en diferentes equipos.
- En Habilitar comunicación SSL, escriba sí.
- En Almacén de claves, introduzca la ubicación del archivo del almacén de claves.
- En Contraseña del almacén de claves, introduzca la contraseña del almacén de claves.
- En Habilitar autenticación SSL del cliente, escriba sí.
- En Almacén de confianza, introduzca la ubicación del archivo del almacén de confianza.
- En Contraseña del almacén de confianza, introduzca la contraseña del almacén de confianza.

El programa de instalación actualiza el archivo de configuración para el agente de extracción. El archivo de configuración se denomina `settings.properties` y se encuentra en la ubicación en la que haya instalado el agente de extracción.

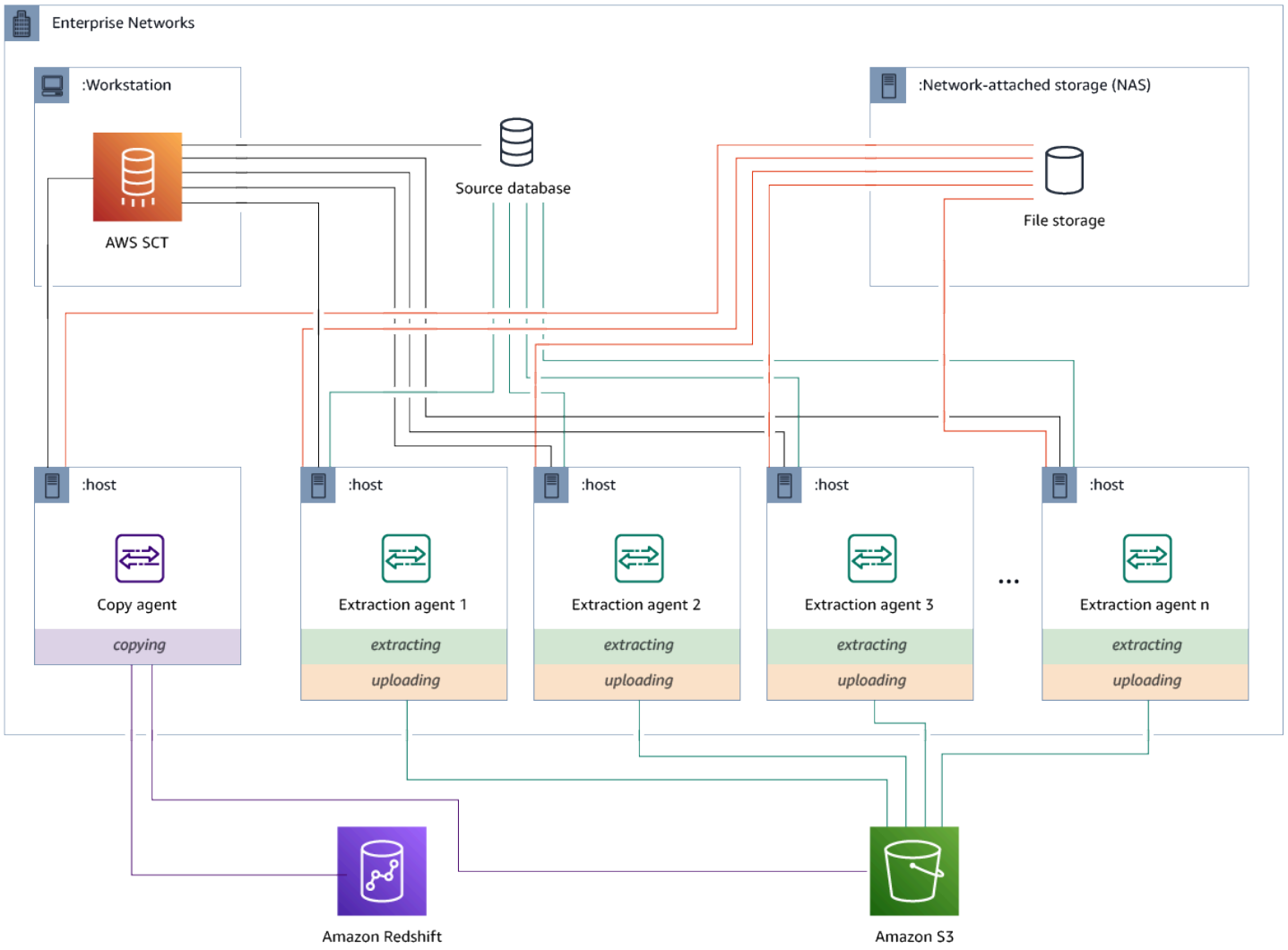
A continuación se muestra un archivo de configuración de ejemplo.

```
$ 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
```

Para cambiar los ajustes de configuración, puede editar el archivo `settings.properties` mediante un editor de texto o volver a ejecutar la configuración del agente.

Instalación y configuración de agentes de extracción con agentes de copia dedicados

Puede instalar los agentes de extracción en una configuración que tenga almacenamiento compartido y un agente de copia dedicado. En el siguiente diagrama se ilustra este escenario.



Esta configuración puede resultar útil cuando un servidor de base de datos de origen admite hasta 120 conexiones y la red tiene un amplio espacio de almacenamiento adjunto. Utilice el siguiente procedimiento para configurar los agentes de extracción que tengan un agente de copia dedicado.

Para instalar y configurar agentes de extracción y un agente de copia dedicado

1. Asegúrese de que el directorio de trabajo de todos los agentes de extracción utilice la misma carpeta en el almacenamiento compartido.

2. Instale los agentes de extracción siguiendo los pasos que se indican en Instalación de agentes de extracción.
3. Configure los agentes de extracción siguiendo los pasos que se indican en Configuración de agentes de extracción, pero especifique únicamente el controlador JDBC de origen.
4. Configure un agente de copia dedicado siguiendo los pasos que se indican en Configuración de agentes de extracción, pero especifique únicamente un controlador JDBC de Amazon Redshift.

Inicio de agentes de extracción

Utilice el siguiente procedimiento para iniciar agentes de extracción. Repita este procedimiento para cada equipo en que tenga instalado un agente de extracción.

Los agentes de extracción actúan en escucha. Cuando inicia un agente con este procedimiento, el agente comienza a escuchar para obtener instrucciones. Enviará a los agentes instrucciones para extraer los datos de su data warehouse en una sección posterior.

Para iniciar su agente de extracción

- En el equipo en el que haya instalado el agente de extracción, ejecute el comando que aparezca a continuación para su sistema operativo.

Sistema operativo	Comando de inicio
Microsoft Windows	Haga doble clic en el archivo de lotes `StartAgent.bat` .
RHEL	Ejecute el siguiente comando en la ruta de la carpeta en la que haya instalado el agente: `sudo initctl start sct-extractor`
Ubuntu Linux	Ejecute el siguiente comando en la ruta de la carpeta en la que haya instalado el agente. Utilice el comando adecuado para su versión de Ubuntu. Ubuntu 14.04: `sudo initctl start sct-extractor` Ubuntu 15.04 y versiones posteriores: `sudo systemctl start sct-extractor`

Para comprobar el estado del agente, ejecute el mismo comando, pero sustituya `start` por `status`.

Para detener un agente, ejecute el mismo comando, pero sustituya `start` por `stop`.

Registrar los agentes de extracción con el AWS Schema Conversion Tool

Usted administra sus agentes de extracción mediante AWS SCT. Los agentes de extracción actúan como oyentes. Cuando reciben instrucciones AWS SCT, extraen datos de su almacén de datos.

Utilice el siguiente procedimiento para registrar los agentes de extracción en su AWS SCT proyecto.

Para registrar un agente de extracción

1. Inicie el AWS Schema Conversion Tool proyecto y ábralo.
2. Abra el menú Ver y seleccione Vista de migración de datos (otra). Aparecerá la pestaña Agentes. Si ya ha registrado agentes, AWS SCT los muestra en una cuadrícula en la parte superior de la pestaña.
3. Elija Registrar.

   Después de registrar a un agente en un AWS SCT proyecto, no podrá registrar al mismo agente en un proyecto diferente. Si ya no utilizas un agente en un AWS SCT proyecto, puedes anular su registro. A continuación, puede registrarlo con otro proyecto diferente.
4. Seleccione Agente de datos de Redshift y, a continuación, elija Aceptar.
5. Introduzca su información en la pestaña Conexión del cuadro de diálogo:

   a. En Descripción, escriba una descripción del agente.
   b. En Nombre de host, escriba el nombre de host o la dirección IP del equipo del agente.
   c. En Puerto, escriba el número de puerto en el que esté escuchando el agente.
   d. Selecciona Registrar para registrar al agente en tu AWS SCT proyecto.
6. Repita los pasos anteriores para registrar múltiples agentes con su proyecto de AWS SCT .

Ocultar y recuperar la información de un AWS SCT agente

Un AWS SCT agente cifra una cantidad significativa de información, por ejemplo, las contraseñas de los almacenes de confianza clave de los usuarios, las cuentas de las bases de datos, la información

de las cuentas y elementos similares. AWS Hace esto utilizando un archivo especial llamado `seed.dat`. De forma predeterminada, el agente crea este archivo en la carpeta de trabajo del usuario que configura el agente por primera vez.

Como diferentes usuarios pueden configurar y ejecutar el agente, la ruta de `seed.dat` se almacena en el parámetro `{extractor.private.folder}` del archivo `settings.properties`. Cuando se inicia el agente, puede utilizar esta ruta para encontrar el archivo `seed.dat` y obtener acceso a la información del almacén de claves y confianza de la base de datos correspondiente.

Es posible que necesite recuperar las contraseñas que un agente ha almacenado en estos casos:

- Si el usuario pierde el `seed.dat` archivo y la ubicación y el puerto del AWS SCT agente no han cambiado.
- Si el usuario pierde el `seed.dat` archivo y la ubicación y el puerto del AWS SCT agente han cambiado. En este caso, el cambio suele ocurrir porque el agente se ha migrado a otro host o puerto y la información del archivo `seed.dat` ya no es válida.

En estos casos, si se inicia un agente sin SSL, este se inicia y después obtiene acceso al almacenamiento del agente creado con anterioridad. A continuación adopta el estado Esperando recuperación.

No obstante, en estos casos, si se inicia un agente con SSL, no puede reiniciarlo. Esto se debe a que el agente no puede descifrar las contraseñas para los certificados almacenados en el archivo `settings.properties`. En este tipo de inicio, el agente no se puede iniciar. Se escribe un error similar al siguiente en el registro: "El agente no se pudo iniciar con el modo SSL habilitado. Vuelva configurar el agente. Motivo: La contraseña de keystore es incorrecta".

Para solucionar este problema, cree un nuevo agente y configúrelo de forma que use las contraseñas existentes para obtener acceso a los certificados SSL. Para ello, siga el procedimiento que se indica a continuación.

Tras realizar este procedimiento, el agente debería ejecutarse y pasar al estado Esperando la recuperación. AWS SCT envía automáticamente las contraseñas necesarias a un agente en estado Esperando la recuperación. Cuando el agente tiene las contraseñas, reinicia todas las tareas. No se requiere ninguna acción adicional por parte de AWS SCT .

Para volver a configurar el agente y restaurar las contraseñas para obtener acceso a los certificados SSL

1. Instale un AWS SCT agente nuevo y ejecute la configuración.
2. Cambie la propiedad `agent.name` del archivo `instance.properties` al nombre del agente para el que se creó el almacenamiento, para que el nuevo agente trabaje con el almacenamiento del agente existente.

   El archivo `instance.properties` se almacena en la carpeta privada del agente, que recibe su nombre siguiendo esta convención: {*`output.folder`*}\dmt\{*`hostName`*}_{*`portNumber`*}\.
3. Cambie el nombre de {*`output.folder`*} a la carpeta de salida del agente anterior.

   En este punto, todavía AWS SCT está intentando acceder al antiguo extractor en el antiguo host y puerto. Por consiguiente, el extractor inaccesible obtiene el estado FAILED. A continuación, puede cambiar el host y el puerto.
4. Modifique el host, el puerto o ambas cosas del agente anterior mediante el comando Modify para redirigir el flujo de solicitudes al nuevo agente.

Cuando AWS SCT puede hacer ping al nuevo agente, AWS SCT recibe el estado Esperando la recuperación del agente. AWS SCT a continuación, recupera automáticamente las contraseñas del agente.

Cada agente que trabaja con el almacenamiento del agente actualiza un archivo especial llamado `storage.lck` ubicado en {*`output.folder`*}\{*`agentName`*}\storage\. Este archivo contiene el ID de red del agente y el tiempo hasta que el almacenamiento está bloqueado. Cuando el agente trabaja con el almacenamiento del agente, actualiza el archivo `storage.lck` y amplía el arrendamiento del almacenamiento en 10 minutos cada 5 minutos. Ninguna otra instancia puede trabajar con el almacenamiento de este agente hasta que finalice el arrendamiento.

Crear reglas de migración de datos en AWS SCT

Antes de extraer los datos con el AWS Schema Conversion Tool, puede configurar filtros que reduzcan la cantidad de datos que extrae. Puede crear reglas de migración de datos mediante el uso de cláusulas `WHERE` para reducir los datos que puede extraer. Por ejemplo, puede escribir una cláusula `WHERE` que seleccione los datos de una única tabla.

Puede crear reglas de migración de datos y guardar los filtros como parte del proyecto. Con el proyecto abierto, utilice el siguiente procedimiento para crear reglas de migración de datos.

Para crear reglas de migración de datos

1. Abra el menú Ver y seleccione Vista de migración de datos (otra).
2. Elija Reglas de migración de datos y, a continuación, elija Agregar nueva regla.
3. Configure su regla de migración de datos:
    a. En Nombre, introduzca un nombre para la regla de migración de datos.
    b. En Donde el nombre del esquema es como, escriba un filtro para aplicárselo a los esquemas. En este filtro, una cláusula `WHERE` se evalúa mediante una cláusula `LIKE`. Para elegir un esquema, introduzca un nombre de esquema exacto. Para elegir varios esquemas, utilice el carácter «%» como comodín para que coincida con cualquier número de caracteres del nombre del esquema.
    c. En Nombre de tabla como, escriba un filtro para aplicárselo a las tablas. En este filtro, una cláusula `WHERE` se evalúa mediante una cláusula `LIKE`. Para elegir una tabla, introduzca un nombre exacto. Para elegir varias tablas, utilice el carácter «%» como comodín para que coincida con cualquier número de caracteres del nombre de la tabla.
    d. En Cláusula Where, escriba una cláusula `WHERE` para filtrar los datos.
4. Una vez que haya configurado su filtro, seleccione Guardar para guardar el filtro o Cancelar para cancelar los cambios.
5. Cuando haya acabado de agregar, editar y eliminar filtros, seleccione Guardar todo para guardar todos los cambios.

Puede utilizar el icono de alternar para desactivar un filtro sin eliminarlo. También puede utilizar el icono de copia para duplicar un filtro existente. Para borrar un filtro existente, utilice el icono de eliminar. Para guardar los cambios que realice en sus filtros, seleccione Guardar todo.

Cambio de los ajustes de extracción y copia en la configuración del proyecto

Desde la ventana de configuración del proyecto AWS SCT, puede elegir la configuración para los agentes de extracción de datos y el comando Amazon Redshift`COPY`.

Para elegir estos ajustes, seleccione Configuración, Configuración del proyecto y, a continuación, Migración de datos. Aquí puede editar los Ajustes de extracción, la Configuración de S3 y los Ajustes de copia.

Siga las instrucciones de la tabla siguiente para proporcionar la información sobre los Ajustes de extracción.

Para este parámetro	Haga lo siguiente
Formato de compresión	Especifique el formato de compresión de los archivos de entrada. Elija una de las siguientes opciones: GZIP, BZIP2, ZSTD o Sin compresión.
Carácter delimitador	Especifique el carácter ASCII que separa los campos de los archivos de entrada. No se admiten caracteres no imprimibles.
Valor NULL como cadena	Active esta opción si los datos incluyen un terminador null. Si esta opción está desactivada, el comando COPY de Amazon Redshift considera null como el final del registro y finaliza el proceso de carga.
Estrategia de ordenación	Utilice la ordenación para reiniciar la extracción desde el punto en el que se produjo el error. Elija una de las siguiente s estrategias de ordenación: Usar la ordenación después del primer fallo (recomendado), Usar la ordenación si es posible o No usar nunca la ordenación. Para obtener más información, consulte the section called “Ordenación de datos”.
Esquema temporal de origen	Introduzca el nombre del esquema en la base de datos de origen, donde el agente de extracción puede crear los objetos temporales.
Tamaño del archivo de salida (en MB)	Introduzca el tamaño, en MB, de los archivos cargados en Amazon S3.
Tamaño del archivo de salida de Snowball (en MB)	Introduzca el tamaño, en MB, de los archivos cargados. AWS Snowball Los archivos pueden tener un tamaño de 1 a 1000 MB.

Para este parámetro	Haga lo siguiente
Usar particionamiento automático. Para Greenplum y Netezza, introducir el tamaño mínimo de las tablas admitidas (en megabytes)	Active esta opción para utilizar particiones en las tablas y, a continuación, introduzca el tamaño de las tablas que desee particionar para las bases de datos fuente de Greenplum y Netezza. Para las migraciones de Oracle a Amazon Redshift, puedes mantener este campo vacío porque AWS SCT crea subtareas para todas las tablas particionadas.
Extraer LOB	Active esta opción para extraer objetos grandes (LOB) de la base de datos de origen. Los LOB incluyen BLOB, CLOB, NCLOB, archivos XML, entre otros. Para cada LOB, los agentes de extracción de AWS SCT crean un archivo de datos.
Carpeta de LOB del bucket de Amazon S3	Introduzca la ubicación de los agentes de AWS SCT extracción para almacenar las LOB.
Aplicar RTRIM a columnas de cadenas	Active esta opción para recortar un conjunto específico de caracteres del final de las cadenas extraídas.
Conservar los archivos de forma local después de subirlos a Amazon S3	Active esta opción para conservar los archivos en su máquina local después de que los agentes de extracción de datos los carguen en Amazon S3.

Siga las instrucciones de la siguiente tabla para proporcionar la información sobre la Configuración de Amazon S3.

Para este parámetro	Haga lo siguiente
Usar proxy	Active esta opción para utilizar un servidor proxy para cargar datos en Amazon S3. Luego elija el protocolo de transferencia de datos e introduzca el nombre del host, el puerto, el nombre de usuario y la contraseña.
Tipo de punto de conexión	Elija FIPS para usar el punto de conexión del estándar federal de procesamiento de información (FIPS). Seleccione VPCE para usar el punto de conexión de la nube privada virtual (VPC).

Para este parámetro	Haga lo siguiente
	Seguidamente, en Punto de conexión de VPC, introduzca el sistema de nombres de dominio (DNS) del punto de conexión de VPC.
Conservar los archivos en Amazon S3 después de copiarlos en Amazon Redshift	Active esta opción para conservar los archivos extraídos en Amazon S3 después de copiarlos en Amazon Redshift.

Siga las instrucciones de la siguiente tabla para proporcionar la información sobre los Ajustes de copia.

Para este parámetro	Haga lo siguiente
Recuento máximo de errores	Introduzca el número de errores de carga. Una vez que la operación alcanza este límite, los agentes de extracción de AWS SCT datos finalizan el proceso de carga de datos. El valor predeterminado es 0, lo que significa que los agentes de extracción de AWS SCT datos continúan con la carga de datos independientemente de los errores.
Sustituir caracteres UTF-8 no válidos	Active esta opción para reemplazar los caracteres UTF-8 no válidos por el carácter especificado y continuar con la operación de carga de datos.
Usar espacio en blanco como valor nulo	Active esta opción para cargar como nulos los campos en blanco que constan de espacios en blanco.
Usar vacío como valor nulo	Active esta opción para cargar como nulos los campos `CHAR` y `VARCHAR` vacíos.
Truncar columnas	Active esta opción para truncar los datos de las columnas de forma que se ajusten a la especificación del tipo de datos.
Compresión automática	Active esta opción para aplicar la codificación de compresión durante una operación de copia.

Para este parámetro	Haga lo siguiente
Actualización automática de estadísticas	Active esta opción para actualizar las estadísticas al final de una operación de copia.
Comprobar archivo antes de cargar	Active esta opción para validar los archivos de datos antes de cargarlos en Amazon Redshift.

Ordenar los datos antes de migrarlos mediante AWS SCT

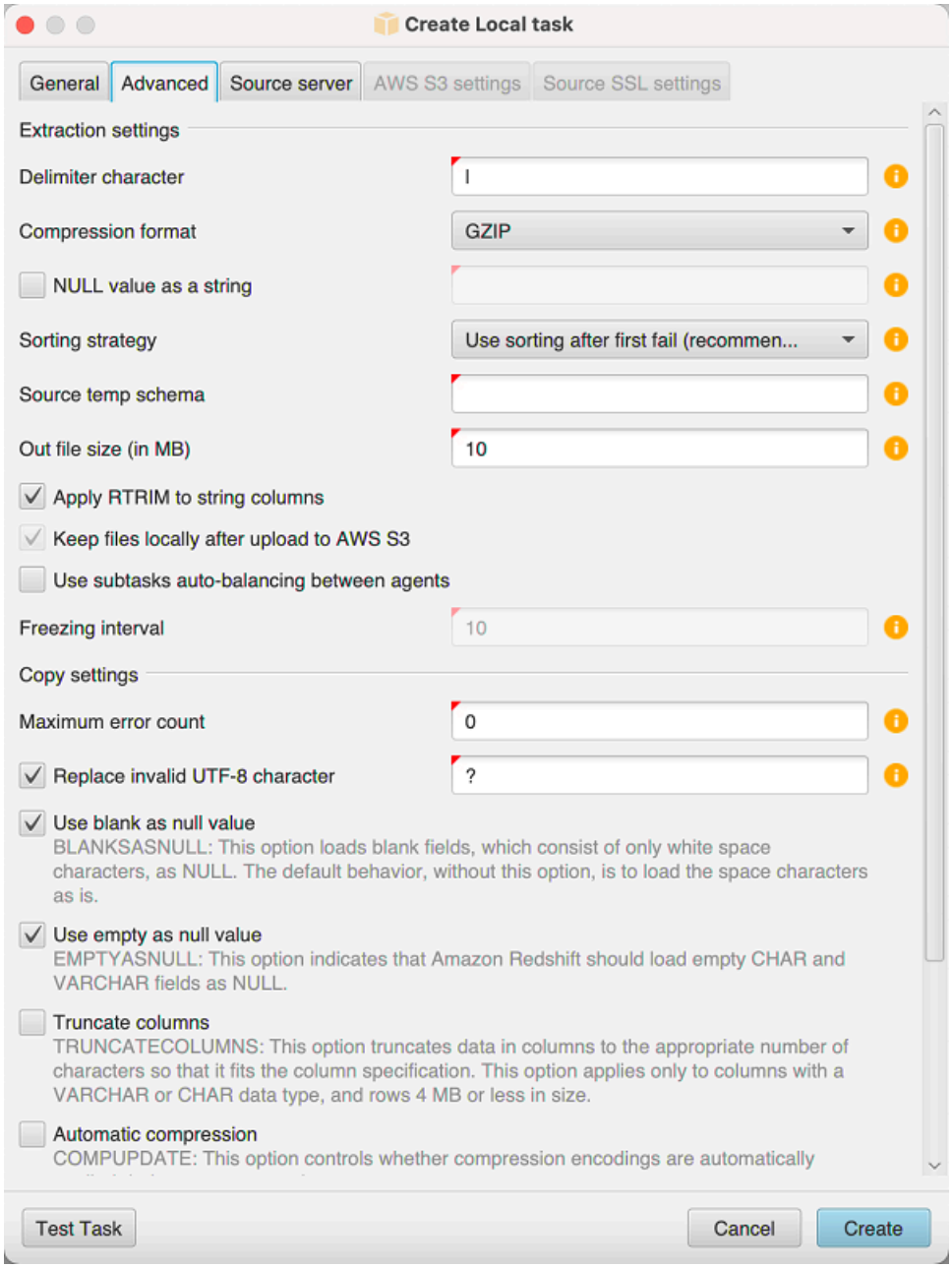
Clasificar los datos antes de la migración AWS SCT ofrece algunas ventajas. Si primero ordena los datos, AWS SCT puede reiniciar el agente de extracción en el último punto guardado después de un error. Además, si va a migrar datos a Amazon Redshift y los ordena primero AWS SCT , podrá insertarlos en Amazon Redshift más rápido.

Estos beneficios tienen que ver con la forma en que AWS SCT crea las consultas de extracción de datos. En algunos casos, AWS SCT utiliza la función analítica DENSE_RANK en estas consultas. Sin embargo, DENSE_RANK puede utilizar mucho tiempo y recursos del servidor para ordenar el conjunto de datos que resulta de la extracción, de modo que si AWS SCT puede funcionar sin él, funciona.

Para ordenar los datos antes de migrarlos utilizando AWS SCT

1. Abra un AWS SCT proyecto.
2. Abra el menú contextual (clic secundario) para el objeto y seleccione Crear tarea local.
3. Seleccione la pestaña Avanzado y elija una opción para Estrategia de ordenación:

    - No usar nunca la ordenación: el agente de extracción no utiliza la función analítica DENSE_RANK y se reinicia desde el principio si se produce un error.
    - Usar la ordenación si es posible: el agente de extracción utiliza DENSE_RANK si la tabla tiene una clave principal o una restricción única.
    - Usar la ordenación después del primer fallo (recomendado): el agente de extracción intenta primero obtener los datos sin usar DENSE_RANK. Si el primer intento falla, el agente de extracción reconstruye la consulta mediante DENSE_RANK y conserva su ubicación en caso de error.



4. Establezca parámetros adicionales, como se describe a continuación, y seleccione Crear para crear la tarea de extracción de datos.

Crear, ejecutar y supervisar una tarea de extracción de AWS SCT datos

Utilice los siguientes procedimientos para crear, ejecutar y supervisar tareas de extracción de datos.

Para asignar tareas a agentes y migrar datos

1. En el panel izquierdo del proyecto AWS Schema Conversion Tool, después de convertir el esquema, elige una o más tablas del panel izquierdo del proyecto.

   Puede seleccionar todas las tablas, pero no se lo recomendamos por motivos de desempeño. Le recomendamos que cree varias tareas para varias tablas basadas en el tamaño de las tablas en su almacenamiento de datos.
2. Abra el menú contextual (clic secundario) para cada tabla y seleccione Crear tarea. Se abre el cuadro de diálogo Crear tarea local.
3. En Nombre de tarea, escriba un nombre para la tarea.
4. En Modo de migración, seleccione una de las siguientes opciones:

   - Solo extraer: extrae los datos y los guarda en sus carpetas de trabajo locales.
   - Extraer y cargar: extrae los datos y los carga en Amazon S3.
   - Extraer, cargar y copiar: extrae los datos, los carga en Amazon S3 y los copia en el almacenamiento de datos de Amazon Redshift.
5. En Tipo de cifrado, seleccione una de las opciones siguientes:

   - NONE: desactive el cifrado de datos durante todo el proceso de migración de datos.
   - CSE_SK: utilice el cifrado del cliente con una clave simétrica para migrar los datos. AWS SCT genera automáticamente claves de cifrado y las transmite a los agentes de extracción de datos por medio de la capa de sockets seguros (SSL). AWS SCT no cifra objetos grandes (LOB) durante la migración de datos.
6. Seleccione Extraer LOB para extraer objetos grandes. Si no es necesario extraer objetos grandes, puede quitar la marca de selección de la casilla. Esto reduce la cantidad de datos que extrae.
7. Si desea ver información detallada sobre una tarea, seleccione Habilitar registro de tareas. Puede utilizar el registro de tareas para depurar problemas.

   Si habilita el registro de tareas, seleccione el nivel de detalle que desea ver. Los niveles son los siguientes y cada nivel incluye todos los mensajes del nivel anterior:

   - ERROR: La cantidad mínima de detalles.
   - WARNING
   - INFO

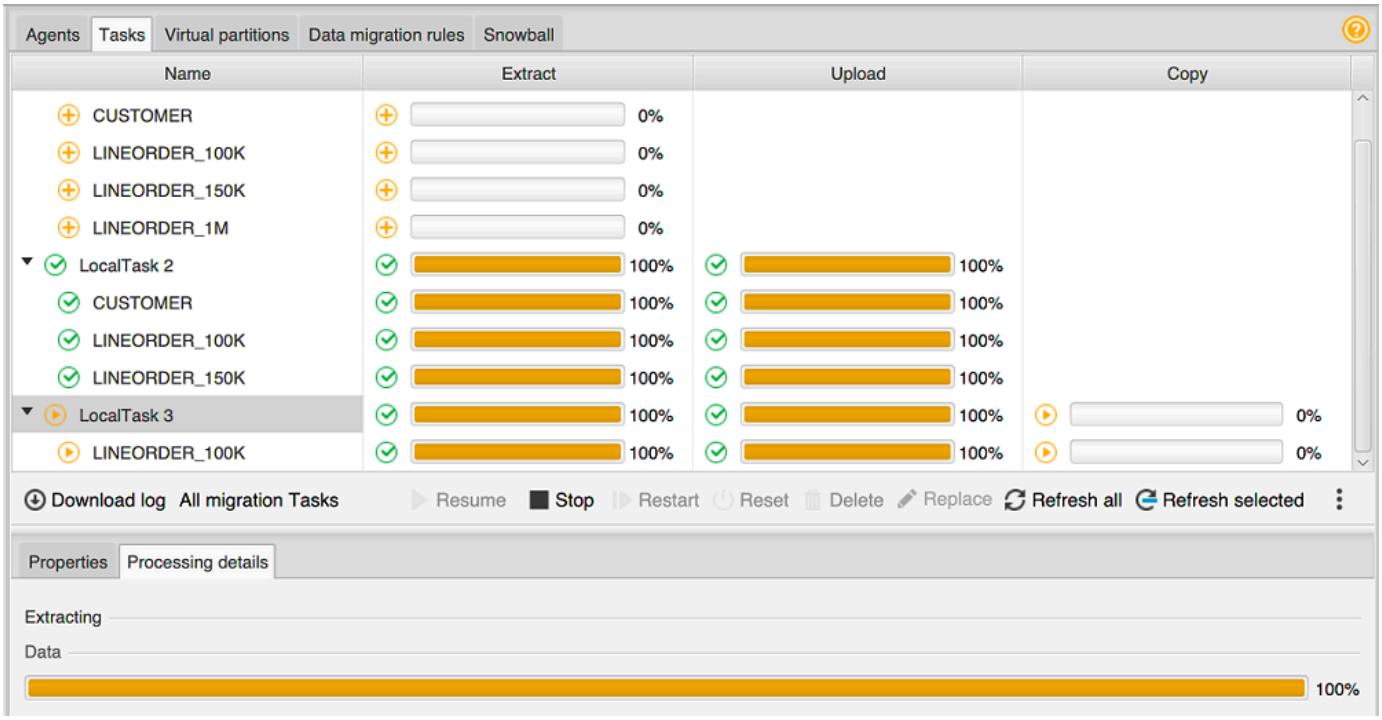
- DEBUG
- TRACE: La cantidad máxima de detalles.

8. Para exportar los datos BigQuery, AWS SCT usa la carpeta bucket de Google Cloud Storage. En esta carpeta, los agentes de extracción de datos almacenan los datos de origen.

   Para introducir la ruta en la carpeta del bucket de Google Cloud Storage, seleccione Avanzado. En Carpeta del bucket de Google CS, introduzca el nombre del bucket y el nombre de la carpeta.
9. Para asumir un rol como usuario del agente de extracción de datos, seleccione Configuración de Amazon S3. En Rol de IAM, introduzca el nombre del rol que desea usar. En Región, elige la Región de AWS para este rol.
10. Seleccione Probar tarea para comprobar que se puede conectar a su carpeta de trabajo, bucket de Amazon S3 y almacenamiento de datos de Amazon Redshift. La verificación depende del modo de migración que elija.
11. Seleccione Crear para crear la tarea.
12. Repita los pasos anteriores para crear tareas para todos los datos que quiera migrar.

Para ejecutar y supervisar tareas

1. En Vista, elija Vista de migración de datos. Aparecerá la pestaña Agentes.
2. Elija la pestaña Tareas. Sus tareas aparecen en la cuadrícula en la parte superior tal y como se muestra a continuación. Podrá ver el estado de una tarea en la cuadrícula superior y el estado de las tareas secundarias en la cuadrícula inferior.



3. Seleccione una tarea en la cuadrícula superior y amplíela. En función del modo de migración que elija, verá la tarea dividida en Extraer, Cargar y Copiar.
4. Para comenzar una tarea, seleccione Iniciar para esa tarea. Puede supervisar el estado de sus tareas mientras están en ejecución. Las tareas secundarias se ejecutan en paralelo. Las funciones de extracción, carga y copia también se ejecutan en paralelo.
5. Si ha habilitado el registro al configurar la tarea, podrá ver el registro:
    a. Seleccione Descargar registro. Aparecerá un mensaje con el nombre de la carpeta que contiene el archivo de registro. Omita el mensaje.
    b. Aparecerá un enlace en la pestaña Detalles de tarea. Seleccione el enlace para abrir la carpeta que contiene el archivo de registro.

Puedes cerrar AWS SCT y tus agentes y tareas seguirán funcionando. Puedes volver a abrirlas AWS SCT más tarde para comprobar el estado de las tareas y ver los registros de las tareas.

Puede guardar las tareas de extracción de datos en el disco local y restaurarlas en el mismo proyecto o en otro proyecto mediante las funciones de exportación e importación. Para exportar una tarea, asegúrese de haber creado al menos una tarea de extracción en un proyecto. Puede importar una sola tarea de extracción o todas las tareas creadas en el proyecto.

Al exportar una tarea de extracción, AWS SCT crea un `.xml` archivo independiente para esa tarea. El archivo `.xml` almacena la información de los metadatos de la tarea, como las propiedades, la descripción y las tareas secundarias de la tarea. El archivo `.xml` no contiene información sobre el procesamiento de una tarea de extracción. Al importar la tarea, se vuelve a crear la siguiente información:

- Progreso de la tarea
- Estados de tareas secundarias y fases
- Distribución de los agentes de extracción por tareas secundarias y fases
- Identificadores de tareas y tareas secundarias
- Nombre de la tarea

Exportación e importación de una tarea AWS SCT de extracción de datos

Puede guardar rápidamente una tarea existente de un proyecto y restaurarla en otro proyecto (o en el mismo proyecto) mediante la AWS SCT exportación y la importación. Utilice el siguiente procedimiento para exportar e importar tareas de extracción de datos.

Para exportar e importar una tarea de extracción de datos

1. En Vista, elija Vista de migración de datos. Aparecerá la pestaña Agentes.
2. Elija la pestaña Tareas. Sus tareas se muestran en la cuadrícula que aparece.
3. Elija los tres puntos alineados verticalmente (icono de puntos suspensivos verticales) situados en la esquina inferior derecha, debajo de la lista de tareas.
4. Seleccione Exportar tarea en el menú emergente.
5. Elija la carpeta en la que desee AWS SCT colocar el `.xml` archivo de exportación de la tarea.

   AWS SCT crea el archivo de exportación de la tarea con un formato de nombre de archivo de*`TASK-DESCRIPTION_TASK-ID`*`.xml`.
6. Elija los tres puntos alineados verticalmente (icono de puntos suspensivos verticales) situados en la esquina inferior derecha, debajo de la lista de tareas.
7. Seleccione Importar tarea en el menú emergente.

Puede importar una tarea de extracción a un proyecto conectado a la base de datos de origen y el proyecto tiene al menos un agente de extracción registrado activo.

8. Seleccione el archivo `.xml` para la tarea de extracción que ha exportado.

   AWS SCT obtiene los parámetros de la tarea de extracción del archivo, crea la tarea y añade la tarea a los agentes de extracción.

9. Repita estos pasos para exportar e importar tareas adicionales de extracción de datos.

Al final de este proceso, la exportación e importación finalizarán y las tareas de extracción de datos estarán listas para su uso.

Extracción de datos mediante un dispositivo AWS Snowball Edge

El proceso de uso AWS SCT de AWS Snowball Edge consta de varios pasos. La migración implica una tarea local, en la que se AWS SCT utiliza un agente de extracción de datos para mover los datos al dispositivo AWS Snowball Edge y, a continuación, una acción intermedia en la que se AWS copian los datos del dispositivo AWS Snowball Edge a un bucket de Amazon S3. El proceso termina AWS SCT de cargar los datos del bucket de Amazon S3 en Amazon Redshift.

Las secciones que siguen a esta descripción general proporcionan una step-by-step guía para cada una de estas tareas. El procedimiento supone que ha AWS SCT instalado, configurado y registrado un agente de extracción de datos en una máquina dedicada.

Realice los siguientes pasos para migrar los datos de un almacén de datos local a un almacén de AWS datos mediante AWS Snowball Edge.

1. Cree un trabajo de AWS Snowball Edge con la AWS Snowball consola.
2. Desbloquee el dispositivo AWS Snowball Edge con la máquina Linux local dedicada.
3. Cree un nuevo proyecto en AWS SCT.
4. Instalar y configurar los agentes de extracción de datos.
5. Crear y establecer permisos para que el bucket de Amazon S3 los utilice.
6. Importa un AWS Snowball trabajo a tu AWS SCT proyecto.
7. Registrar el agente de extracción de datos en AWS SCT.
8. Cree una tarea local en AWS SCT.

9. Ejecutar y supervisar la tarea de migración de datos en AWS SCT.

S: tep-by-step procedimientos para migrar datos mediante un AWS SCT Edge AWS Snowball

En las secciones siguientes se incluye información detallada sobre los pasos para la migración.

Paso 1: Crear un trabajo de AWS Snowball Edge

Cree un AWS Snowball trabajo siguiendo los pasos descritos en la sección Creación de un trabajo de AWS Snowball Edge de la Guía para desarrolladores de AWS Snowball Edge.

Paso 2: desbloquea el dispositivo AWS Snowball Edge

Ejecute los comandos que desbloquean y proporcionan credenciales al dispositivo Snowball Edge desde el equipo en el que instaló el AWS DMS agente. Al ejecutar estos comandos, puede estar seguro de que la llamada al AWS DMS agente se conecta al dispositivo AWS Snowball Edge. Para obtener más información sobre cómo desbloquear el dispositivo AWS Snowball Edge, consulte Desbloquear Snowball Edge.

```
aws s3 ls s3://<bucket-name> --profile <Snowball Edge profile> --endpoint http://
<Snowball IP>:8080 --recursive
```

Paso 3: Crea un nuevo proyecto AWS SCT

A continuación, crea un AWS SCT proyecto nuevo.

Para crear un proyecto nuevo en AWS SCT

1. Inicie el AWS Schema Conversion Tool. En el menú Archivo, seleccione Proyecto nuevo. Aparece el cuadro de diálogo Proyecto nuevo.
2. Introduzca un nombre para su proyecto, que se almacenará localmente en su equipo.
3. Introduzca la ubicación del archivo local del proyecto.
4. Pulse Aceptar para crear su AWS SCT proyecto.
5. Seleccione Añadir fuente para añadir una nueva base de datos fuente a su AWS SCT proyecto.
6. Seleccione Añadir destino para añadir una nueva plataforma de destino a su AWS SCT proyecto.

7. Elija el esquema de la base de datos de origen en el panel izquierdo.
8. En el panel derecho, especifique la plataforma de la base de datos de destino para el esquema de origen seleccionado.
9. Seleccione Crear asignación. Este botón se activa después de elegir el esquema de la base de datos de origen y la plataforma de la base de datos de destino.

Paso 4: Instalar y configurar su agente de extracción de datos

AWS SCT utiliza un agente de extracción de datos para migrar los datos a Amazon Redshift. El archivo.zip que descargó para la instalación incluye el archivo AWS SCT instalador del agente de extracción. Puede instalar el agente de extracción de datos en Windows, Red Hat Enterprise Linux o Ubuntu. Para obtener más información, consulte Instalación de agentes de extracción.

Para configurar el agente de extracción de datos, introduzca los motores de base de datos de origen y destino. Además, asegúrese de haber descargado los controladores JDBC para las bases de datos de origen y destino en el equipo en el que ejecuta el agente de extracción de datos. Los agentes de extracción de datos utilizan estos controladores para conectarse a las bases de datos de origen y de destino. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios.

En Windows, el instalador del agente de extracción de datos inicia el asistente de configuración en la ventana de símbolo del sistema. En Linux, ejecute el archivo `sct-extractor-setup.sh` desde la ubicación en la que instaló el agente.

Paso 5: Configurar el acceso AWS SCT al bucket de Amazon S3

Para obtener más información acerca la configuración de un bucket de Amazon S3, consulte Descripción general de los buckets en la Guía del usuario de Amazon Simple Storage Service.

Paso 6: Importa un AWS Snowball trabajo a tu AWS SCT proyecto

Para conectar el AWS SCT proyecto con el dispositivo AWS Snowball Edge, importe el AWS Snowball trabajo.

Para importar tu AWS Snowball trabajo

1. Abra el menú Configuración y seleccione Configuración global. Aparecerá el cuadro de diálogo Configuración global.

2. Seleccione Perfiles de servicios de AWS y, a continuación, seleccione Importar trabajo.
3. Elige tu AWS Snowball trabajo.
4. Introduzca el IP de AWS Snowball . Para obtener más información, consulte Cambio de la dirección IP en la Guía del usuario de AWS Snowball .
5. Introduce tu AWS Snowball puerto. Para obtener más información, consulte los puertos necesarios para usar AWS los servicios en un dispositivo AWS Snowball Edge en la Guía para desarrolladores de AWS Snowball Edge.
6. Especifique su clave de acceso de AWS Snowball y su clave secreta de AWS Snowball . Para obtener más información, consulte Autenticación y control de acceso en AWS Snowball en la Guía del usuario de AWS Snowball .
7. Elija Aplicar y, después, Aceptar.

Paso 7: Registre un agente de extracción de datos en AWS SCT

En esta sección, registrará el agente de extracción de datos en AWS SCT.

Para registrar un agente de extracción de datos

1. En el menú Ver, seleccione Vista de migración de datos (otra) y, seguidamente, seleccione Registrar.
2. En Descripción, introduzca un nombre para el agente de extracción de datos.
3. En Nombre de host, introduzca la dirección IP del equipo en el que ejecuta el agente de extracción de datos.
4. En Puerto, introduzca el puerto oyente que configuró previamente.
5. Elija Registro.

Paso 8: Crear una tarea local

A continuación, cree la tarea de migración. La tarea incluye dos tareas secundarias. Una subtarea migra los datos de la base de datos de origen al dispositivo AWS Snowball Edge. La otra tarea secundaria toma los datos que el dispositivo cara a un bucket de Amazon S3 y los migra a la base de datos de destino.

Para crear una tarea de migración

1. En el menú Ver, seleccione Vista de migración de datos (otra).

2. En el panel de la izquierda que muestra el esquema de la base de datos de origen, seleccione un objeto del esquema para su migración. Abra el menú contextual (clic secundario) para el objeto y seleccione Crear tarea local.
3. En Nombre de la tarea, introduzca un nombre para la tarea de migración de datos.
4. En Modo de migración, seleccione Extraer, cargar y copiar.
5. Seleccione Configuración de Amazon S3.
6. Seleccione Usar Snowball.
7. Introduzca las carpetas y subcarpetas en el bucket de Amazon S3 donde el agente de extracción de datos pueda almacenar los datos.
8. Seleccione Crear para crear la tarea.

Paso 9: Ejecutar y monitorear la tarea de migración de datos en AWS SCT

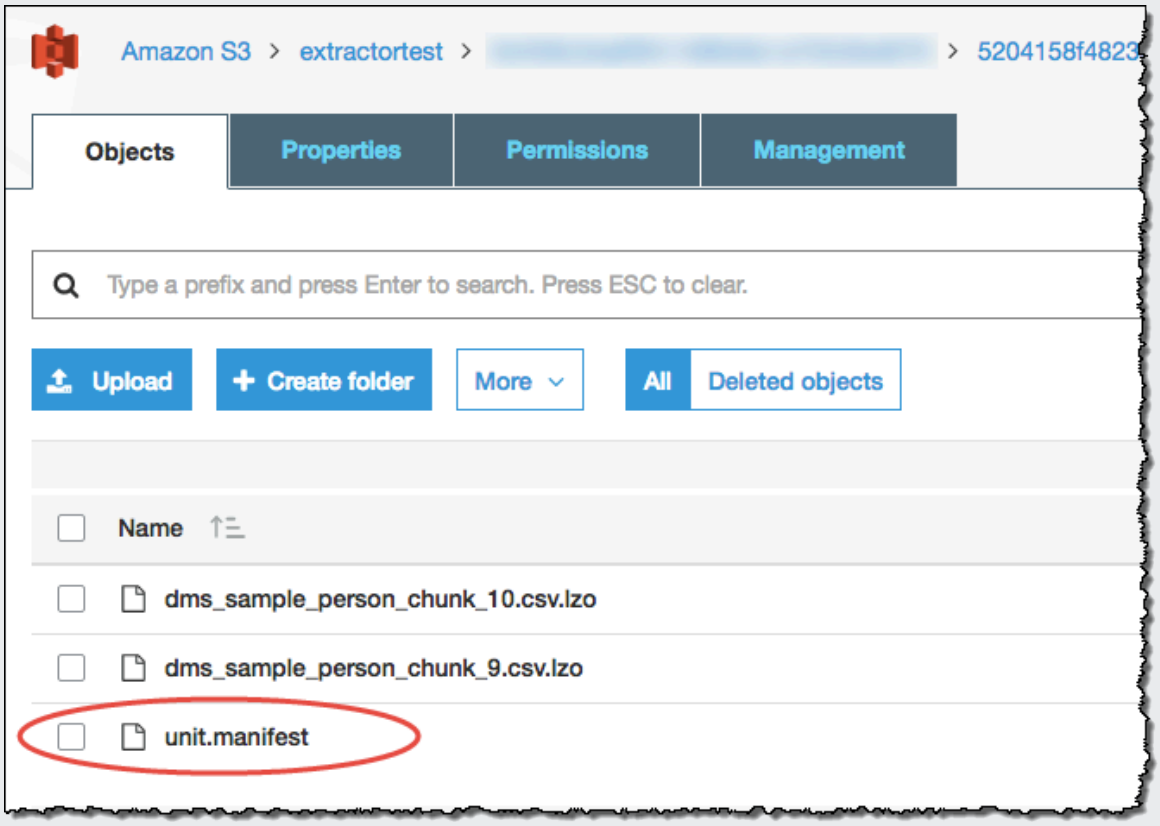
Para iniciar la tarea de migración de datos, elija Iniciar. Asegúrese de haber establecido conexiones con la base de datos de origen, el bucket de Amazon S3, el AWS Snowball dispositivo y la conexión con la base de datos de destino en ella AWS.

Puede supervisar y administrar las tareas de migración de datos y sus tareas secundarias en la pestaña Tareas. Puede ver el progreso de la migración de datos, así como pausar o reiniciar las tareas de migración de datos.

Resultado de la tarea de extracción de datos

Tras completarse sus tareas de migración, sus datos estarán listos. Utilice la siguiente información para determinar cómo proceder en función del modo de migración que haya elegido y la ubicación de los datos.

Modo de migración	Ubicación de los datos
Extraer, cargar y copiar	Los datos ya están en el almacenamiento de datos de Amazon Redshift. Puede verificar que los datos están ahí y empezar a utilizarlos. Para obtener más información, consulte la sección Conectar a clústeres desde herramientas cliente y código.

Modo de migración	Ubicación de los datos
Extraer y cargar	Los agentes de extracción guardaron sus datos como archivos en el bucket de Amazon S3. Puede usar el comando COPY de Amazon Redshift para cargar los datos en Amazon Redshift. Para obtener más información, consulte Cargar datos desde Amazon S3 en la documentación de Amazon Redshift. En el bucket de Amazon S3 hay varias carpetas, que se corresponden con las tareas de extracción que configuró. Al cargar sus datos en Amazon Redshift, especifique el nombre del archivo de manifiesto creado por cada tarea. El archivo de manifiesto aparece en la carpeta de tareas del bucket de Amazon S3, tal y como se muestra a continuación. 
Solo extraer	Los agentes de extracción guardaron sus datos en su carpeta de trabajo. Copie a mano sus datos en el bucket de Amazon S3; y, a continuación, proceda con las instrucciones para Extraer y cargar.

Uso de particiones virtuales con AWS Schema Conversion Tool

A menudo, podrá administrar mejor grandes tablas sin particiones mediante la creación de tareas secundarias que crean particiones virtuales de los datos de la tabla utilizando reglas de filtrado. En AWS SCT, puede crear particiones virtuales para los datos migrados. Existen tres tipos de partición, que funcionan con determinados tipos de datos:

- El tipo de partición RANGE funciona con tipos de datos de fecha y hora y numéricos.
- El tipo de partición LIST funciona con tipos de datos de fecha y hora, numéricos y caracteres.
- El tipo de partición DATE AUTO SPLIT funciona con tipos de datos numéricos, de fecha y de hora.

AWS SCT valida los valores que proporciona para crear una partición. Por ejemplo, si intenta particionar una columna con el tipo de datos NUMERIC pero proporciona valores de un tipo de datos diferente, AWS SCT arroja un error.

Además, si va AWS SCT a migrar datos a Amazon Redshift, puede utilizar la partición nativa para gestionar la migración de tablas grandes. Para obtener más información, consulte Uso de particiones nativas.

Límites del particionamiento virtual

Estas son las limitaciones al crear una partición virtual:

- Solo puede utilizar particionamiento virtual para tablas no particionadas.
- Solo puede utilizar particionamiento virtual en la vista de migración de datos.
- No puede utilizar la opción UNION ALL VIEW con particionamiento virtual.

Tipo de partición RANGE

El tipo de partición RANGE realiza particiones de datos basadas en un intervalo de valores de la columna para tipos de datos de fecha y hora y numéricos. Este tipo de partición crea una cláusula `WHERE` y usted proporciona el intervalo de valores de cada partición. Para especificar una lista de valores para la columna particionada, utilice el cuadro Valores. Puede cargar información del valor mediante un archivo .csv.

El tipo de partición RANGE crea particiones predeterminadas en ambos extremos de los valores de las particiones. Estas particiones predeterminadas capturan cualquier dato inferior o superior a los valores de partición especificados.

Por ejemplo, puede crear varias particiones basadas en un intervalo de valores que proporcione. En el siguiente ejemplo, se especifican los valores de particionamiento para LO_TAX para crear varias particiones.

```
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
```

Para crear una partición virtual RANGE

1. Abrir. AWS SCT
2. Elija el modo Vista de migración de datos (otra).
3. Elija la tabla en la que desea configurar el particionamiento virtual. Abra el menú contextual (clic secundario) de la tabla y seleccione Agregar particionamiento virtual.
4. En el cuadro de diálogo Agregar particionamiento virtual, introduzca la información, como sigue.

Opción	Acción
Tipo de partición	Seleccione RANGE. La interfaz de usuario del cuadro de diálogo cambia en función del tipo que elija.
Nombre de la columna	Seleccione la columna en la que desea realizar la partición.
Tipo de columna	Elija el tipo de datos para los valores en la columna.
Valores	Para agregar nuevos valores, escriba cada valor en el cuadro Valor nuevo y, a continuación, elija el signo más para agregar el valor.
Cargar desde archivo	(Opcional) Escriba el nombre de un archivo .csv que contenga valores de partición.

5. Seleccione Aceptar.

Tipo de partición LIST

El tipo de partición LIST realiza particiones de datos basadas en los valores de la columna para tipos de datos de fecha y hora, numéricos y caracteres. Este tipo de partición crea una cláusula `WHERE` y usted proporciona los valores de cada partición. Para especificar una lista de valores para la columna particionada, utilice el cuadro Valores. Puede cargar información del valor mediante un archivo .csv.

Por ejemplo, puede crear varias particiones basadas en un valor que proporcione. En el siguiente ejemplo, se especifican los valores de particionamiento para LO_ORDERKEY para crear varias particiones.

```
Partition1: WHERE LO_ORDERKEY = 1
Partition2: WHERE LO_ORDERKEY = 2
Partition3: WHERE LO_ORDERKEY = 3
…
PartitionN: WHERE LO_ORDERKEY = USER_VALUE_N
```

También puede crear una partición predeterminada para valores no incluidos en los especificados.

Puede usar el tipo de partición LIST para filtrar los datos de origen si desea excluir determinados valores de la migración. Por ejemplo, suponga que desea omitir filas con `LO_ORDERKEY = 4`. En este caso, no incluya el valor `4` en la lista de valores de partición y asegúrese de que no se elija la opción Incluir otros valores.

Para crear una partición virtual LIST

1. Abrir AWS SCT.
2. Elija el modo Vista de migración de datos (otra).
3. Elija la tabla en la que desea configurar el particionamiento virtual. Abra el menú contextual (clic secundario) de la tabla y seleccione Agregar particionamiento virtual.
4. En el cuadro de diálogo Agregar particionamiento virtual, introduzca la información, como sigue.

Opción	Acción
Tipo de partición	Elija LIST. La interfaz de usuario del cuadro de diálogo cambia en función del tipo que elija.

Opción	Acción
Nombre de la columna	Seleccione la columna en la que desea realizar la partición.
Valor nuevo	Aquí puede escribir un valor que se agregará al conjunto de valores de particionamiento.
Incluir otros valores	Seleccione esta opción para crear una partición predeterminada en la que se almacenan todos los valores que no cumplen los criterios de particionamiento.
Cargar desde archivo	(Opcional) Escriba el nombre de un archivo .csv que contenga valores de partición.

5. Seleccione Aceptar.

Tipo de partición DATE AUTO SPLIT

El tipo de partición DATE AUTO SPLIT es una forma automática de generar particiones RANGE. Con DATA AUTO SPLIT, usted AWS SCT indica el atributo de partición, dónde empezar y dónde terminar, y el tamaño del rango entre los valores. A continuación, AWS SCT calcula los valores de la partición automáticamente.

DATA AUTO SPLIT automatiza gran parte del trabajo que implica la creación de particiones de rango. El equilibrio entre el uso de esta técnica y el particionamiento de rango es el grado de control que se necesita sobre los límites de las particiones. El proceso de división automática siempre crea rangos de igual tamaño (uniformes). El particionamiento de rango le permite variar el tamaño de cada rango según sea necesario para su distribución de datos particular. Por ejemplo, puede usar diariamente, semanalmente, quincenalmente, mensualmente, entre otros.

```
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’
```

Para crear una partición virtual DATE AUTO SPLIT

1. Abrir AWS SCT.
2. Elija el modo Vista de migración de datos (otra).
3. Elija la tabla en la que desea configurar el particionamiento virtual. Abra el menú contextual (clic secundario) de la tabla y seleccione Agregar particionamiento virtual.
4. En el cuadro de diálogo Agregar particionamiento virtual, introduzca la siguiente información.

Opción	Acción
Tipo de partición	Seleccione DATE AUTO SPLIT. La interfaz de usuario del cuadro de diálogo cambia en función del tipo que elija.
Nombre de la columna	Seleccione la columna en la que desea realizar la partición.
Fecha de inicio	Escriba una fecha de inicio.
Fecha de finalización	Escriba una fecha de finalización.
Intervalo	Escriba la unidad del intervalo y elija el valor de dicha unidad.

5. Seleccione Aceptar.

Uso de particiones nativas

Para acelerar la migración de datos, sus agentes de extracción de datos pueden usar particiones nativas de tablas en el servidor de almacenamiento de datos de origen. AWS SCT admite el particionamiento nativo para las migraciones de Greenplum, Netezza y Oracle a Amazon Redshift.

Por ejemplo, después de crear un proyecto, puede recopilar estadísticas en un esquema y analizar el tamaño de las tablas seleccionadas para la migración. En el caso de las tablas que superen el tamaño especificado, se activa el mecanismo de particionamiento nativo AWS SCT .

Para usar particionamiento nativo

1. Abra AWS SCT y elija Nuevo proyecto para Archivo. Aparece el cuadro de diálogo Proyecto nuevo.

2. Cree un proyecto nuevo, añada los servidores de origen y destino, y cree reglas de asignación. Para obtener más información, consulte Creación de un proyecto de AWS SCT.
3. Haga clic en Ver y, a continuación, seleccione Vista principal.
4. En Configuración del proyecto, elija la pestaña Migración de datos. Seleccione Usar particiones automáticas. Para las bases de datos de origen de Greenplum y Netezza, introduzca el tamaño mínimo de las tablas admitidas en megabytes (por ejemplo, 100). AWS SCT crea automáticamente tareas secundarias de migración independientes para cada partición nativa que no esté vacía. Para las migraciones de Oracle a Amazon Redshift, AWS SCT crea subtareas para todas las tablas particionadas.
5. En el panel de la izquierda que muestra el esquema de la base de datos de origen, seleccione un esquema. Abra el menú contextual (clic secundario) para el objeto y seleccione Recopilar estadísticas. Para migrar datos desde Oracle a Amazon Redshift, puede omitir este paso.
6. Elija todas las tablas que desee migrar.
7. Registre el número necesario de agentes. Para obtener más información, consulte Registrar los agentes de extracción con el AWS Schema Conversion Tool.
8. Cree una tarea de extracción de datos para las tablas seleccionadas. Para obtener más información, consulte Crear, ejecutar y supervisar una tarea de extracción de AWS SCT datos.

   Compruebe si las tablas grandes se dividen en tareas secundarias y si cada tarea secundaria coincide con el conjunto de datos que presenta una parte de la tabla ubicada en un sector del almacenamiento de datos de origen.
9. Inicie y supervise el proceso de migración hasta que AWS SCT los agentes de extracción de datos completen la migración de los datos de sus tablas de origen.

Migración de los LOB a Amazon Redshift

Amazon Redshift no admite el almacenamiento de objetos binarios grandes (LOB). Sin embargo, si necesita migrar una o más LOB a Amazon Redshift AWS SCT , puede realizar la migración. Para ello, AWS SCT utiliza un bucket de Amazon S3 para almacenar los LOB y escribe la dirección URL para el bucket de Amazon S3 en los datos migrados almacenados en Amazon Redshift.

Para migrar LOB a Amazon Redshift

1. Abra un AWS SCT proyecto.
2. Conéctese a bases de datos de origen y destino. Actualice los metadatos de la base de datos de destino y asegúrese de que las tablas convertidas estén allí.

3. En Acciones, seleccione Crear tarea local.
4. En Modo de migración, seleccione una de las siguientes opciones:
   - Extraer y cargar para extraer sus datos y los cargarlos en Amazon S3.
   - Extraer, cargar y copiar para extraer sus datos, cargarlos en Amazon S3 y copiarlos en el almacenamiento de datos de Amazon Redshift.
5. Seleccione Configuración de Amazon S3.
6. En Carpeta de LOB del bucket de Amazon S3, escriba el nombre de la carpeta de un bucket de S3 donde desee almacenar los LOB.

   Si utiliza el perfil de AWS servicio, este campo es opcional. AWS SCT puede usar la configuración predeterminada de su perfil. Para usar otro bucket de Amazon S3, introduzca la ruta aquí.
7. Active la opción Usar proxy para utilizar un servidor proxy para cargar datos en Amazon S3. Luego elija el protocolo de transferencia de datos e introduzca el nombre del host, el puerto, el nombre de usuario y la contraseña.
8. En Tipo de punto de conexión, seleccione FIPS para usar el punto de conexión del estándar federal de procesamiento de información (FIPS). Seleccione VPCE para usar el punto de conexión de la nube privada virtual (VPC). Seguidamente, en Punto de conexión de VPC, introduzca el sistema de nombres de dominio (DNS) del punto de conexión de VPC.
9. Active la opción Conservar archivos en Amazon S3 después de copiarlos en Amazon Redshift para conservar los archivos extraídos en Amazon S3 después de copiarlos en Amazon Redshift.
10. Seleccione Crear para crear la tarea.

Prácticas recomendadas y solución de problemas de agentes de extracción de datos

A continuación se indican algunas prácticas recomendadas y sugerencias de resolución de problemas para el uso de los agentes de extracción.

Problema	Sugerencias para la solución de problemas
El desempeño es lento	Para mejorar el desempeño, le recomendamos lo siguiente: • Instale varios agentes.

Problema	Sugerencias para la solución de problemas
	• Instale agentes en equipos cercanos a su almacenamiento de datos. • No ejecute todas las tablas en una única tarea de agente.
Evite los retrasos	Evite tener demasiados agentes que obtengan acceso a su almacenamiento de datos al mismo tiempo.
Un agente deja de funcionar de forma temporal	Si un agente deja de funcionar, el estado de cada una de sus tareas aparecerá como erróneo en la AWS SCT. En algunos casos, el agente puede recuperarse con solo esperar. En este caso, el estado de sus tareas se actualiza en la AWS SCT.
Un agente deja de funcionar de forma permanente	Si el equipo que ejecuta un agente deja de funcionar de forma permanente, y ese agente está ejecutando una tarea, puede sustituirlo por un nuevo agente que continúe con la tarea. Solo puede sustituir un nuevo agente si la carpeta de trabajo del agente original no se encuentra en el mismo equipo que el agente original. Para sustituir un nuevo agente, haga lo siguiente: • Instale un agente en un nuevo equipo. • Configure el nuevo agente con los mismos ajustes, incluido el número de puerto y la carpeta de trabajo, que el agente original. • Comience el agente. Tras iniciar el agente, la tarea descubrir á al nuevo agente disponible y seguirá ejecutándose en el nuevo agente.

Conversión de SQL de las aplicaciones mediante AWS SCT

Al convertir su esquema de base de datos de un motor a otro, también deberá actualizar el código SQL de las aplicaciones para interactuar con el nuevo motor de base de datos en lugar del antiguo. Puede ver, analizar, editar y guardar el código SQL convertido.

Puede utilizar AWS Schema Conversion Tool (AWS SCT) para convertir el código SQL en código C++, C #, Java u otro tipo de código de su aplicación. En el caso de una conversión de Oracle a PostgreSQL, puede utilizar AWS SCT para convertir el código de SQL*Plus a PSQL. Además, para una conversión de Oracle a PostgreSQL, puede utilizar AWS SCT para convertir el código SQL incrustado en aplicaciones de C#, C++, Java y Pro*C.

Temas

- Información general de la conversión del SQL de las aplicaciones
- Conversión del código SQL de las aplicaciones con AWS SCT
- Conversión del código SQL de aplicaciones de C# con AWS SCT
- Convertir el código SQL de aplicaciones de C++ con AWS SCT
- Conversión del código SQL en aplicaciones de Java con AWS SCT
- Conversión del código SQL en aplicaciones de Pro*C con AWS SCT

Información general de la conversión del SQL de las aplicaciones

Para convertir el código SQL de su aplicación, siga estos pasos de alto nivel:

- Cree un proyecto de conversión de la aplicaciones: el proyecto de conversión de la aplicación es secundario al proyecto de conversión del esquema de la base de datos. Cada proyecto de conversión de esquema de base de datos puede tener uno o varios proyectos de conversión de la aplicación secundarios. Para obtener más información, consulte Creación de proyectos de conversión genérica de aplicaciones en AWS SCT.
- Analice y convierta su código SQL: AWS SCT analiza su aplicación, extrae el código SQL y crea una versión local del SQL convertido para que pueda analizarlo y editarlo. La herramienta no cambia el código en su aplicación hasta que esté listo. Para obtener más información, consulte Analizar y convertir su código SQL en AWS SCT.
- Cree un informe de evaluación de la aplicación: el informe de evaluación de la aplicación proporciona información importante sobre la conversión del código SQL de la aplicación desde el

esquema de la base de datos de origen al esquema de la base de datos de destino. Para obtener más información, consulte Creación y uso del informe de evaluación de AWS SCT en AWS SCT.

- Edite, aplique los cambios y guarde el código SQL convertido: el informe de evaluación incluye una lista de elementos de código SQL que no se pueden convertir automáticamente. Para estos elementos, puede editar el código SQL manualmente para realizar la conversión. Para obtener más información, consulte Editar y guardar el código SQL convertido con AWS SCT.

Conversión del código SQL de las aplicaciones con AWS SCT

Puede utilizar AWS SCT para convertir el código SQL incrustado en sus aplicaciones. El conversor genérico de aplicaciones de AWS SCT trata el código de la aplicación como texto sin formato. Analiza el código de la aplicación y extrae el código SQL con expresiones regulares. Este conversor admite diferentes tipos de archivos de código fuente y funciona con el código de la aplicación escrito en cualquier lenguaje de programación.

El conversor genérico de aplicaciones tiene las siguientes limitaciones. No profundiza en la lógica de la aplicación que es específica del lenguaje de programación de la aplicación. Además, el conversor genérico no admite sentencias SQL de diferentes objetos de la aplicación, como funciones, parámetros, variables locales, etc.

Para mejorar la conversión del código SQL de la aplicación, utilice conversores de código SQL de aplicaciones específicos del lenguaje correspondiente. Para obtener más información, consulte Convertir el código SQL de aplicaciones de C#, Convertir el código SQL en aplicaciones de Java y Conversión del código SQL en aplicaciones de Pro*C.

Creación de proyectos de conversión genérica de aplicaciones en AWS SCT

En AWS Schema Conversion Tool, el proyecto de conversión de la aplicación es secundario al proyecto de conversión del esquema de la base de datos. Cada proyecto de conversión de esquema de base de datos puede tener uno o varios proyectos de conversión de la aplicación secundarios.

> **Note**
>
> AWS SCT no admite la conversión entre los siguientes orígenes y destinos:
>
> - Oracle a Oracle
> - PostgreSQL a PostgreSQL o Aurora PostgreSQL

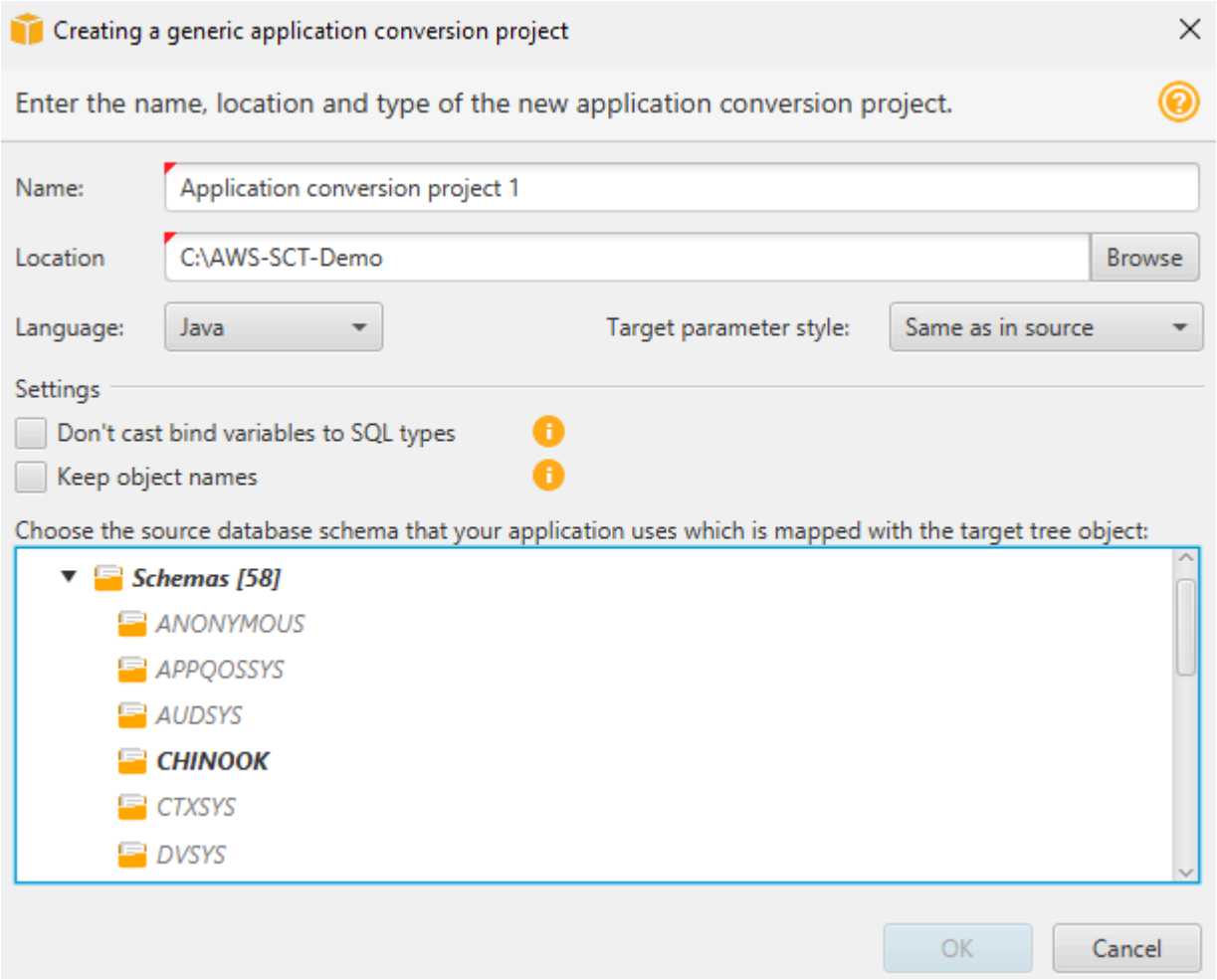
- MySQL a mysql
- SQL Server a SQL Server
- Amazon Redshift a Amazon Redshift
- SQL Server a Babelfish
- SQL Server Integration Services a AWS Glue
- Apache Cassandra a Amazon DynamoDB

Utilice el siguiente procedimiento para crear un proyecto de conversión genérica de aplicación.

Para crear un proyecto de conversión de aplicación

1. En AWS Schema Conversion Tool, seleccione Aplicación nueva desde el menú Aplicaciones.

   Aparecerá el cuadro de diálogo Proyecto de conversión de aplicación nuevo.



2. Agregue la siguiente información del proyecto.

Para este parámetro	Haga lo siguiente
Nombre	Escriba un nombre para el proyecto de conversión de la aplicación. Cada proyecto de conversión de esquemas de bases de datos puede tener uno o varios proyectos secundarios de conversión de aplicaciones, por lo que deberá elegir un nombre significativo si va a agregar más proyectos posteriormente.
Ubicación	Escriba la ubicación del código fuente de su aplicación.
Lenguaje	Elija una de las siguientes opciones: • Java • C++ • C# • Cualquiera
Estilo de parámetro de destino	Elija la sintaxis que se utilizará para vincular variables en el código convertido. Las distintas plataformas de bases de datos utilizan una sintaxis diferente para las variables de enlace. Elija una de las siguientes opciones: • Same as in source • Positional (?) • Indexed (:1) • Indexed ($1) • Named (@name) • Named (:name) • Named (&name) • Named ($name) • Named (#name) • Named (!name!)

Para este parámetro	Haga lo siguiente
Elija el esquema de la base de datos de origen	En el árbol de origen, elija el esquema utilizado por la aplicación. Asegúrese de que este esquema forma parte de una regla de asignación.

3. Seleccione No convertir las variables de enlace en tipos de SQL para evitar la conversión de los tipos de variables de enlace en tipos de SQL. Esta opción solo está disponible para una conversión de Oracle a PostgreSQL.

   Por ejemplo, el código fuente de la aplicación incluye la siguiente consulta de Oracle:

   ```
   SELECT * FROM ACCOUNT WHERE id = ?
   ```

   Si selecciona No convertir las variables de enlace en tipos de SQL, AWS SCT convierte esta consulta como se muestra a continuación.

   ```
   SELECT * FROM account WHERE id = ?
   ```

   Si desactiva la opción No convertir las variables de enlace en tipos de SQL, AWS SCT cambia el tipo de variable de enlace por el tipo de datos `NUMERIC`. El resultado de la conversión se muestra a continuación.

   ```
   SELECT * FROM account WHERE id = (?)::NUMERIC
   ```

4. Seleccione Conservar nombres de objetos para evitar agregar el nombre del esquema al nombre del objeto convertido. Esta opción solo está disponible para una conversión de Oracle a PostgreSQL.

   Por ejemplo, imagino que el código fuente de la aplicación incluye la siguiente consulta de Oracle.

   ```
   SELECT * FROM ACCOUNT
   ```

   Al seleccionar Conservar nombres de objetos, AWS SCT convierte esta consulta como se muestra a continuación.

   ```
   SELECT * FROM account
   ```

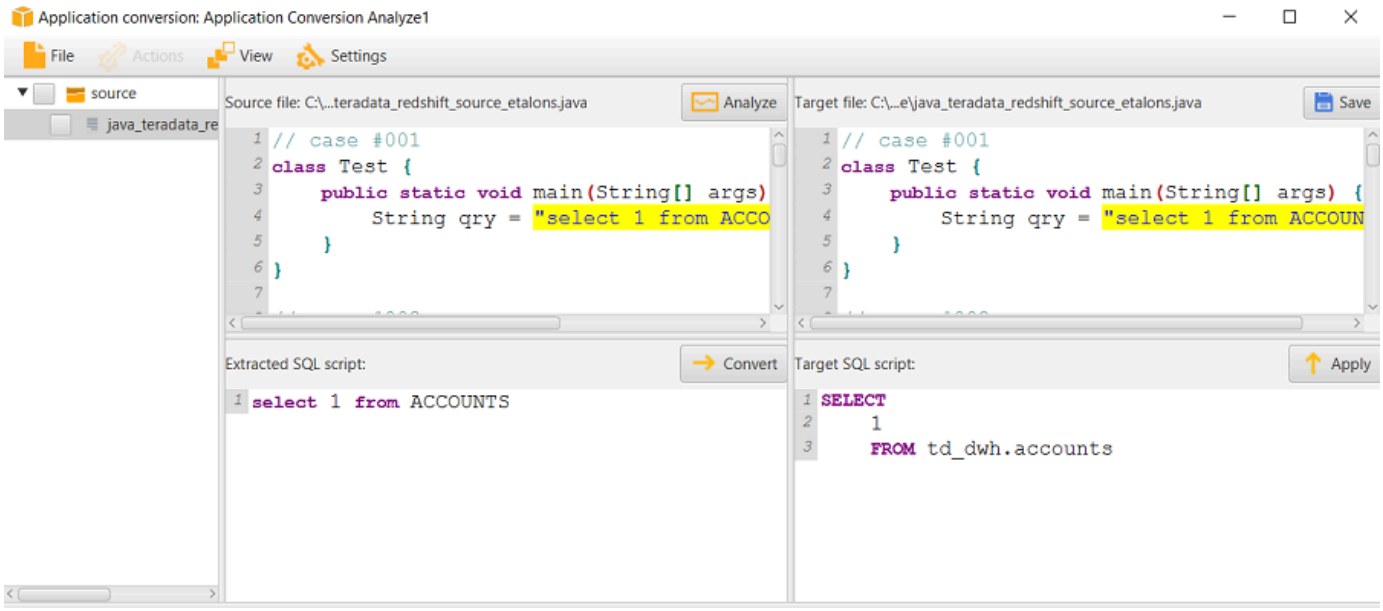
Si desactiva Conservar nombres de objetos, AWS SCT agrega el nombre del esquema al nombre de la tabla. El resultado de la conversión se muestra a continuación.

```
SELECT * FROM schema_name.account
```

Si el código fuente incluye los nombres de los objetos principales en los nombres de los objetos, AWS SCT utiliza este formato en el código convertido. En este caso, ignore la opción Conservar nombres de objetos porque AWS SCT agrega los nombres de los objetos principales al código convertido.

5. Seleccione Aceptar para crear su proyecto de conversión de la aplicación.

   Se abrirá la ventana de proyecto.



Administrar proyectos de conversión de aplicaciones en AWS SCT

Puede abrir un proyecto de conversión de aplicaciones existente y agregar varios proyectos de conversión de aplicaciones.

Despues de crear un proyecto de conversión de aplicaciones, la ventana de proyecto se abrirá automáticamente. Puede cerrar la ventana del proyecto de conversión de aplicaciones y volver a ella más adelante.

Para crear un proyecto de conversión de aplicaciones

1. En el panel izquierdo, elija el nodo del proyecto de conversión de aplicaciones y abra el menú contextual (clic secundario).
2. Elija Administrar aplicación.

Para agregar un proyecto de conversión de aplicaciones adicional

1. En el panel izquierdo, elija el nodo del proyecto de conversión de aplicaciones y abra el menú contextual (clic secundario).
2. Elija Aplicación nueva.
3. Introduzca la información necesaria para crear un proyecto de conversión de aplicaciones nuevo. Para obtener más información, consulte Crear proyectos de conversión genérica de aplicaciones.

Analizar y convertir su código SQL en AWS SCT

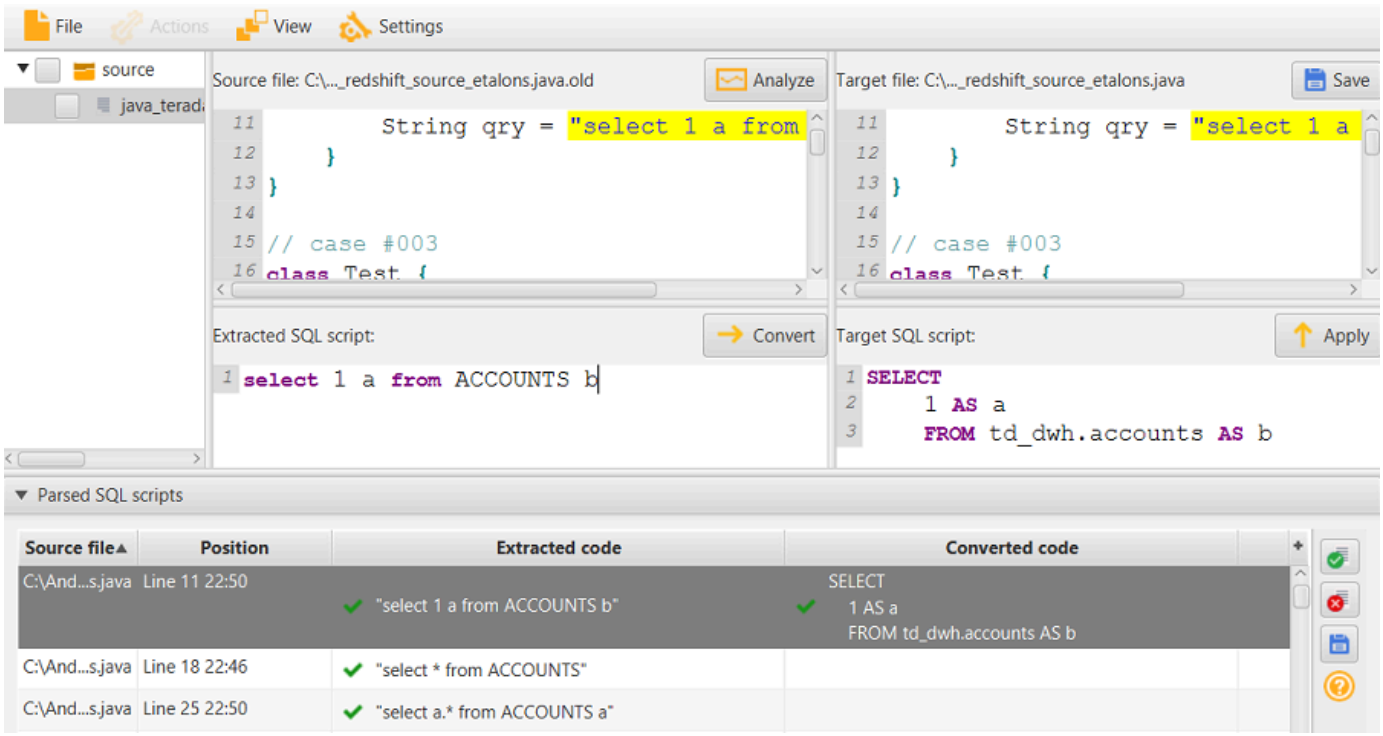
Utilice el siguiente procedimiento para analizar y convertir su código SQL en AWS Schema Conversion Tool.

Para analizar y convertir su código SQL

1. Abra un proyecto de conversión de aplicaciones existente y elija Analizar.

   AWS SCT analiza el código de la aplicación y extrae el código SQL. AWS SCT muestra el código SQL extraído en la lista Scripts de SQL analizados.
2. En Scripts de SQL analizados, elija un elemento para revisar el código SQL extraído. AWS SCT muestra el código del elemento seleccionado en el panel Scripts de SQL extraídos.
3. Seleccione Convertir para convertir el código SQL en el panel Scripts de SQL extraído. AWS SCT convierte el código a un formato compatible con la base de datos de destino.

   Puede editar el código SQL convertido. Para obtener más información, consulte Editar y guardar el código SQL convertido.



4. Al crear un informe de evaluación de la conversión de una aplicación, AWS SCT convierte todos los elementos de código SQL extraídos. Para obtener más información, consulte Crear y usar el informe de evaluación.

Creación y uso del informe de evaluación de AWS SCT en AWS SCT

El informe de evaluación de la conversión de la aplicación proporciona información sobre cómo convertir el código SQL de la aplicación a un formato compatible con la base de datos de destino. En el informe se detalla todo el código SQL extraído de la aplicación, todo el SQL convertido y los elementos de acción del código SQL que AWS SCT no puede convertir.

Creación de un informe de evaluación de la conversión de la aplicación

Utilice el siguiente procedimiento para crear un informe de evaluación de la conversión de la aplicación.

Para crear un informe de evaluación de la conversión de la aplicación

1. En la ventana del proyecto de conversión de la aplicación, seleccione Crear informe en el menú Acciones.

AWS SCT crea el informe de evaluación de la conversión de la aplicación y lo abre en la ventana del proyecto de conversión de la aplicación.

2. Consulte la pestaña Resumen.

   En la pestaña Resumen que se muestra a continuación aparece la información resumida del informe de evaluación de la aplicación. Muestra los elementos del código SQL que convirtieron automáticamente y los elementos que no se convirtieron automáticamente.

3. Seleccione Acciones de extracción de SQL.

   Revise la lista de elementos de código SQL que AWS SCT no puede extraer de su código fuente.

4. Seleccione Acciones de conversión de SQL.

   Revise la lista de elementos de código SQL que AWS SCT no puede convertir automáticamente. Utilice las acciones recomendadas para convertir manualmente el código SQL. Para obtener información acerca de cómo editar el código SQL convertido, consulte Editar y guardar el código SQL convertido con AWS SCT.

5. (Opcional) Guarde una copia local del informe como archivo PDF o como un archivo de valores separados por comas (CSV):

   - Seleccione Guardar en PDF en la parte superior derecha para guardar el informe como un archivo PDF.

     El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de la aplicación.

   - Seleccione Guardar en CSV en la parte superior derecha para guardar el informe como un archivo CSV.

     El archivo CSV contiene elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir el código SQL.

Editar y guardar el código SQL convertido con AWS SCT

El informe de evaluación incluye una lista de elementos de código SQL que AWS SCT no puede convertir automáticamente. Para cada elemento, AWS SCT crea un elemento de acción en la pestaña Acciones de conversión de SQL. Para estos elementos, puede editar el código SQL manualmente para realizar la conversión.

Utilice el siguiente procedimiento para editar su código SQL convertido, aplique los cambios y, a continuación, guárdelos.

Para editar, aplicar cambios y guardar su código SQL convertido

1. Edite su código SQL convertido directamente en el panel Target SQL script. Si no se muestra ningún código convertido, puede hacer clic en el panel y empezar a escribir.
2. Una vez que haya terminado de editar su código SQL convertido, seleccione Aplicar. En este punto, los cambios se guardan en memoria, pero aún no se escriben en el archivo.
3. Seleccione Guardar para guardar los cambios en su archivo.

   Al seleccionar Guardar, sobrescribirá su archivo original. Realice una copia de su archivo original antes de guardar, a fin de que disponga de un registro de su código de aplicación original.

Conversión del código SQL de aplicaciones de C# con AWS SCT

Para una conversión de Oracle a PostgreSQL, puede utilizar AWS Schema Conversion Tool (AWS SCT) para convertir el código SQL incrustado en sus aplicaciones de C#. Este conversor de aplicaciones de C# específico comprende la lógica de la aplicación. Recopila las declaraciones que se encuentran en diferentes objetos de la aplicación, como funciones, parámetros, variables locales, etc.

Gracias a este análisis profundo, el conversor de código SQL de aplicaciones de C# proporciona mejores resultados de conversión que el conversor genérico.

Crear proyectos de conversión de aplicaciones de C# en AWS SCT

Puede crear un proyecto de conversión de aplicaciones de C# solo para convertir esquemas de bases de datos Oracle en esquemas de bases de datos PostgreSQL. Agregue una regla de asignación a su proyecto que incluya un esquema de Oracle de origen y una base de datos

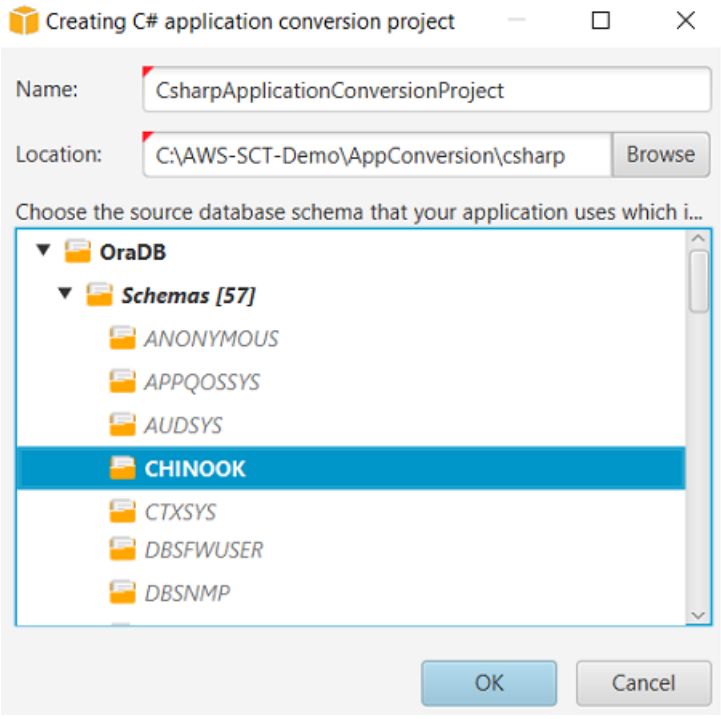
PostgreSQL de destino. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT.

Puede agregar varios proyectos de conversión de aplicaciones en un único AWS SCT proyecto. Utilice el siguiente procedimiento para crear un proyecto de conversión de aplicaciones de C#.

Para crear un proyecto de conversión de aplicaciones de C#

1. Cree un proyecto de conversión de bases de datos y agregue una base de datos Oracle de origen. Para obtener más información, consulte Creación de un proyecto de AWS SCT y Cómo agregar servidores de bases de datos a un proyecto de AWS SCT.
2. Agregue una regla de asignación que incluya la base de datos Oracle de origen y una base de datos PostgreSQL de destino. Puede agregar una base de datos PostgreSQL de destino o utilizar una plataforma de base de datos PostgreSQL de destino virtual en una regla de asignación. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT y Usar destinos virtuales.
3. En el menú Ver, seleccione Vista principal.
4. En la página Aplicaciones, elija Crear aplicación.

   Aparecerá el cuadro de diálogo Proyecto de conversión de aplicación de C# nuevo.



5. En Nombre, escriba un nombre para el proyecto de conversión de la aplicación de C#. Puesto que cada proyecto de conversión de esquemas de bases de datos puede tener uno o varios

proyectos secundarios de conversión de aplicaciones, elija un nombre significativo si va agrega varios proyectos.

6. En Ubicación, escriba la ubicación del código fuente de su aplicación.
7. En el árbol de origen, elija el esquema utilizado por la aplicación. Asegúrese de que este esquema forme parte de una regla de asignación. AWS SCT resalta en negrita los esquemas que forman parte de una regla de asignación.
8. Seleccione Aceptar para crear su proyecto de conversión de la aplicación de C#.
9. Busque su proyecto de conversión de la aplicación de C# en el nodo Aplicaciones del panel izquierdo.

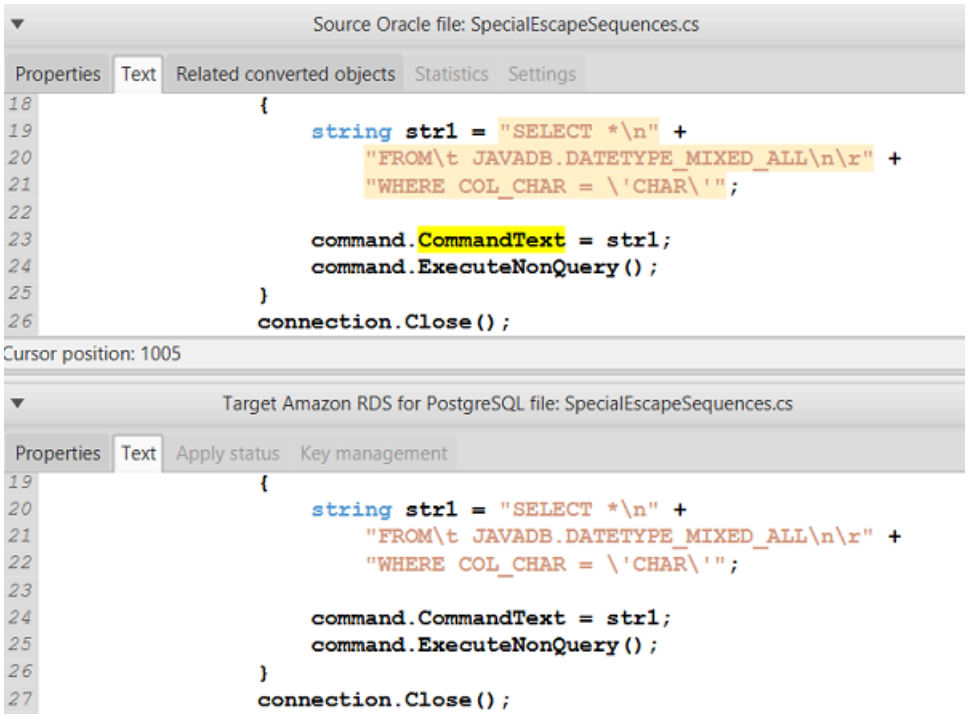
Conversión del código SQL de la aplicación de C# en AWS SCT

Tras agregar la aplicación de C# al proyecto de AWS SCT, convierta el código SQL de esta aplicación a un formato compatible con la plataforma de base de datos de destino. Utilice el siguiente procedimiento para analizar y convertir su código SQL incrustado en su aplicación de C# en AWS Schema Conversion Tool.

Para convertir el código SQL

1. Expanda el nodo C# en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea convertir y abra el menú contextual (clic secundario).
3. Seleccione Convertir. AWS SCT analiza los archivos de código fuente, determina la lógica de la aplicación y carga los metadatos del código en el proyecto. Estos metadatos de código incluyen clases, objetos, métodos, variables globales, interfaces, etc. de C#.

   En el panel de la base de datos de destino, AWS SCT crea una estructura de carpetas similar a la del proyecto de la aplicación de origen. Aquí puede revisar el código de la aplicación convertida.



4. Guarde el código de la aplicación convertida. Para obtener más información, consulte Cómo guardar el código de la aplicación convertida.

Sus aplicaciones de C# pueden incluir código SQL que interactúa con diferentes bases de datos de origen. Puede migrar a PostgreSQL varias de estas bases de datos de origen. En este caso, asegúrese de no convertir el código SQL que interactúa con las bases de datos que excluyó del ámbito de la migración. Puede excluir los archivos de origen de la aplicación de C# del ámbito de la conversión. Para ello, desactive las casillas de verificación de los nombres de los archivos que desee excluir del ámbito de la conversión.

Tras modificar el ámbito de la conversión, AWS SCT seguirá analizando el código SQL de todos los archivos de origen de sus aplicaciones de C#. A continuación, AWS SCT copia en la carpeta de destino todos los archivos de origen que excluyó del ámbito de la conversión. Esta operación permite compilar la aplicación después de guardar los archivos de la aplicación convertidos.

Cómo guardar el código de la aplicación convertida con AWS SCT

Utilice el siguiente procedimiento para guardar el código de la aplicación convertida.

Para guardar el código de la aplicación convertida

1. Expanda el nodo C# en Aplicaciones en el panel de la base de datos de destino.

2. Elija la aplicación convertida y elija Guardar.
3. Introduzca la ruta a la carpeta para guardar el código de la aplicación convertida y seleccione Seleccionar carpeta.

Administración de proyectos de conversión de aplicaciones de C# en AWS SCT

Puede agregar varios proyectos de conversión de aplicaciones de C#, actualizar el código de la aplicación del proyecto de AWS SCT o eliminar un proyecto de conversión de aplicaciones de C# del proyecto de AWS SCT.

Para agregar un proyecto de conversión de aplicaciones de C# adicional

1. Expanda el nodo Aplicaciones en el panel izquierdo.
2. Seleccione el nodo C# y abra el menú contextual (clic secundario).
3. Elija Aplicación nueva.
4. Introduzca la información necesaria para crear un proyecto de conversión de aplicaciones de C# nuevo. Para obtener más información, consulte Crear proyectos de conversión de aplicaciones de C#.

Tras realizar los cambios en el código fuente de la aplicación, cárguelo en el proyecto de AWS SCT.

Para actualizar el código de la aplicación actualizado

1. Expanda el nodo C# en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea actualizar y abra el menú contextual (clic secundario).
3. Seleccione Actualizar y, a continuación, elija Sí.

   AWS SCT carga el código de la aplicación desde los archivos de origen y elimina los resultados de la conversión. Para conservar los cambios de código que ha realizado en AWS SCT y los resultados de la conversión, cree un proyecto de conversión de aplicaciones de C# nuevo.

Para eliminar un proyecto de conversión de aplicaciones de C#

1. Expanda el nodo C# en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea eliminar y abra el menú contextual (clic secundario).

3. Elija Eliminar y a continuación haga clic en Aceptar.

Creación de un informe de evaluación de la conversión de la aplicación de C# en AWS SCT

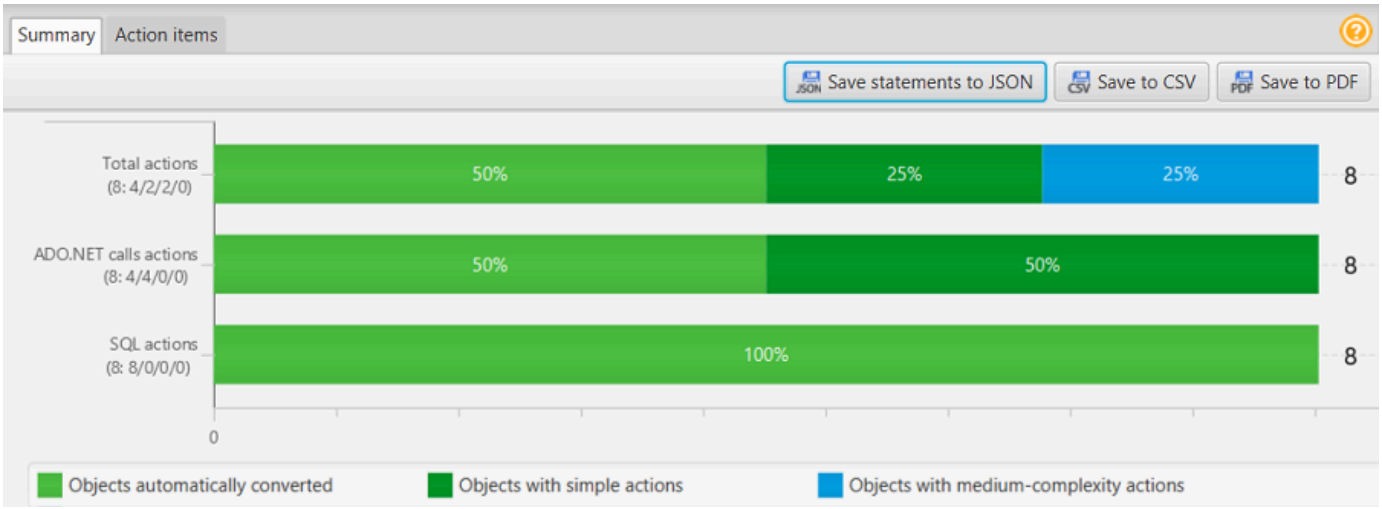
El Informe de evaluación de la conversión de la aplicación de C# proporciona información sobre cómo convertir el código SQL incrustado en su aplicación de C# a un formato compatible con la base de datos de destino. El informe de evaluación proporciona detalles de la conversión para todos los puntos de ejecución de SQL y todos los archivos de código fuente. El informe de evaluación también incluye una lista de elementos de código SQL que AWS SCT no puede convertir.

Utilice el siguiente procedimiento para crear un informe de evaluación de la conversión de una aplicación de C#.

Para crear un informe de evaluación de la conversión de la aplicación de C#

1. Expanda el nodo C# en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea convertir y abra el menú contextual (clic secundario).
3. Seleccione Convertir.
4. En el menú Ver, seleccione Ver informe de evaluación.
5. Consulte la pestaña Resumen.

   En la pestaña Resumen que se muestra a continuación aparece la información del resumen ejecutivo del informe de evaluación de la aplicación de C#. Muestra los resultados de la conversión de todos los puntos de ejecución de SQL y de todos los archivos de código fuente.



6. Seleccione Guardar instrucciones en JSON para guardar el código SQL extraído de la aplicación de C# como un archivo JSON.
7. (Opcional) Guarde una copia local del informe como archivo PDF o como un archivo de valores separados por comas (CSV):

   - Seleccione Guardar en PDF en la parte superior derecha para guardar el informe como un archivo PDF.

     El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de la aplicación.
   - Seleccione Guardar en CSV en la parte superior derecha para guardar el informe como un archivo CSV.

     El archivo CSV contiene elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir el código SQL.

Convertir el código SQL de aplicaciones de C++ con AWS SCT

Para una conversión de Oracle a PostgreSQL, puede utilizar AWS SCT para convertir el código SQL incrustado en sus aplicaciones de C++. Este conversor de aplicaciones de C++ específico comprende la lógica de la aplicación. Recopila las declaraciones que se encuentran en diferentes objetos de la aplicación, como funciones, parámetros, variables locales, etc.

Gracias a este análisis profundo, el conversor de código SQL de aplicaciones de C++ proporciona mejores resultados de conversión que el conversor genérico.

Crear proyectos de conversión de aplicaciones de C++ en AWS SCT

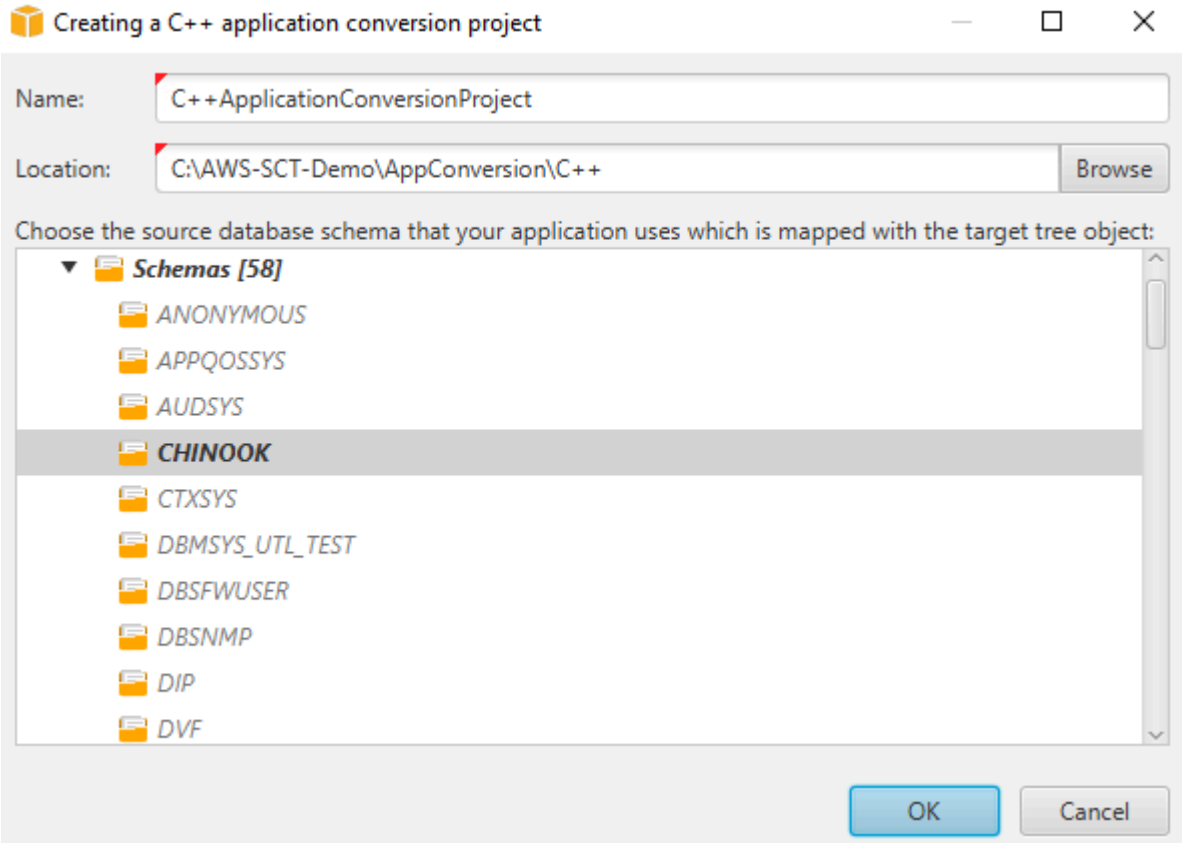
Puede crear un proyecto de conversión de aplicaciones de C++ solo para convertir esquemas de bases de datos Oracle en esquemas de bases de datos PostgreSQL. Agregue una regla de asignación a su proyecto que incluya un esquema de Oracle de origen y una base de datos PostgreSQL de destino. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT.

Puede agregar varios proyectos de conversión de aplicaciones en un único AWS SCT proyecto.

Para crear un proyecto de conversión de aplicaciones de C++

1. Cree un proyecto de conversión de bases de datos y agregue una base de datos Oracle de origen. Para obtener más información, consulte Creación de un proyecto de AWS SCT y Cómo agregar servidores de bases de datos a un proyecto de AWS SCT.
2. Agregue una regla de asignación que incluya la base de datos Oracle de origen y una base de datos PostgreSQL de destino. Puede agregar una base de datos PostgreSQL de destino o utilizar una plataforma de base de datos PostgreSQL de destino virtual en una regla de asignación. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT y Usar destinos virtuales.
3. En el menú Ver, seleccione Vista principal.
4. En el menú Aplicaciones, elija Aplicación de ++ nueva.

   Aparecerá el cuadro de diálogo Proyecto de conversión de aplicación de C++ nuevo.



5. En Nombre, escriba un nombre para el proyecto de conversión de la aplicación de C++. Puesto que cada proyecto de conversión de esquemas de bases de datos puede tener uno o varios proyectos secundarios de conversión de aplicaciones, elija un nombre significativo si va agrega varios proyectos.

6. En Ubicación, escriba la ubicación del código fuente de su aplicación.
7. En el árbol de origen, elija el esquema utilizado por la aplicación. Asegúrese de que este esquema forme parte de una regla de asignación. AWS SCT resalta en negrita los esquemas que forman parte de una regla de asignación.
8. Seleccione Aceptar para crear su proyecto de conversión de la aplicación de C++.
9. Busque su proyecto de conversión de aplicaciones en C++ en el nodo Aplicaciones del panel izquierdo.

Convertir el código SQL de la aplicación de C++ en AWS SCT

Tras agregar la aplicación de C++ al proyecto de AWS SCT, convierta el código SQL de esta aplicación a un formato compatible con la plataforma de base de datos de destino. Utilice el siguiente procedimiento para analizar y convertir su código SQL incrustado en su aplicación de C++ en AWS SCT.

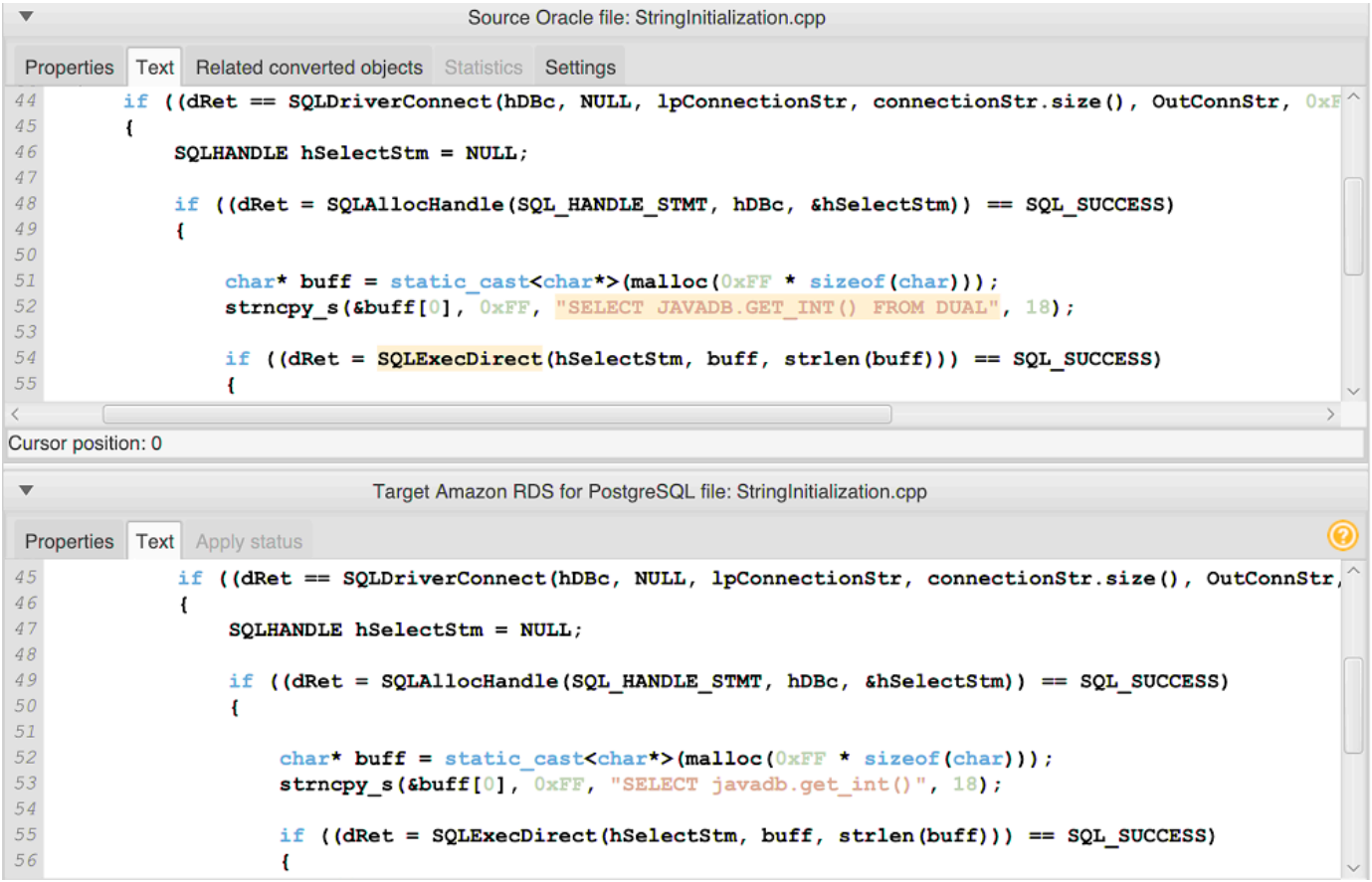
Para convertir el código SQL

1. Expanda el nodo C++ en Aplicaciones en el panel izquierdo y elija la aplicación que desee convertir.
2. En Proyecto de la aplicación Oracle de origen, seleccione Configuración. Revise y edite la configuración de conversión de la aplicación de C++ seleccionada. También puede especificar la configuración de conversión para todas las aplicaciones de C++ que haya agregado al proyecto de AWS SCT. Para obtener más información, consulte Administrar proyectos de conversión de aplicaciones de C++.
3. En Tipo de compilador, elija el compilador que utilizará para el código fuente de la aplicación de C++. AWS SCT admite los siguientes compiladores de C++: Microsoft Visual C++, GCC, GNU Compiler Collection y Clang. La opción predeterminada es Microsoft Visual C++.
4. En Macros definidas por el usuario, introduzca la ruta al archivo que incluye las macros definidas por el usuario del proyecto de C++. Asegúrese de que este archivo tiene la siguiente estructura: `#define name value`. Al igual que en el ejemplo anterior, el parámetro `value` es opcional. El valor predeterminado para este parámetro es 1.

   Para crear este archivo, abra el proyecto en Microsoft Visual Studio y, a continuación, elija Proyecto, Propiedades, C/C++ y Preprocesador. En Definiciones del preprocesador, elija Editar y copie los nombres y valores en un nuevo archivo de texto. A continuación, para cada cadena del archivo, agregue el siguiente prefijo: `#define` .

5. En Directorios de inclusión externa, introduzca las rutas a las carpetas que incluyen las bibliotecas externas que utiliza en su proyecto de C++.
6. En el panel izquierdo, seleccione la aplicación que desea convertir y abra el menú contextual (clic secundario).
7. Seleccione Convertir. AWS SCT analiza los archivos de código fuente, determina la lógica de la aplicación y carga los metadatos del código en el proyecto. Estos metadatos de código incluyen clases, objetos, métodos, variables globales, interfaces, etc. de C++.

   En el panel de la base de datos de destino, AWS SCT crea una estructura de carpetas similar a la del proyecto de la aplicación de origen. Aquí puede revisar el código de la aplicación convertida, como se muestra a continuación.



8. Guarde el código de la aplicación convertida. Para obtener más información, consulte Cómo guardar el código de la aplicación convertida.

Cómo guardar el código de la aplicación convertida con AWS SCT

Utilice el siguiente procedimiento para guardar el código de la aplicación convertida.

Para guardar el código de la aplicación convertida

1. Expanda el nodo C++ en Aplicaciones en el panel de la base de datos de destino.
2. Elija la aplicación convertida y elija Guardar.
3. Introduzca la ruta a la carpeta para guardar el código de la aplicación convertida y seleccione Seleccionar carpeta.

Administrar proyectos de conversión de aplicaciones de C++ en AWS SCT

Puede agregar varios proyectos de conversión de aplicaciones de C++, editar la configuración de la conversión, actualizar el código de la aplicación de C++ o eliminar un proyecto de conversión de aplicaciones de C++ del proyecto de AWS SCT.

Para agregar un proyecto de conversión de aplicaciones de C++ adicional

1. Expanda el nodo Aplicaciones en el panel izquierdo.
2. Seleccione el nodo C++ y abra el menú contextual (clic secundario).
3. Elija Aplicación nueva.
4. Introduzca la información necesaria para crear un proyecto de conversión de aplicaciones de C++ nuevo. Para obtener más información, consulte Crear proyectos de conversión de aplicaciones de C++.

Puede especificar la configuración de la conversión para todos los proyectos de conversión de aplicaciones de C++ del proyecto de AWS SCT.

Para editar la configuración de conversión de todas las aplicaciones de C++

1. En el menú Configuración, elija Configuración del proyecto y, a continuación, elija Conversión de aplicación.
2. En Tipo de compilador, elija el compilador que utilizará para el código fuente de la aplicación de C++. AWS SCT admite los siguientes compiladores de C++: Microsoft Visual C++, GCC, GNU Compiler Collection y Clang. La opción predeterminada es Microsoft Visual C++.
3. En Macros definidas por el usuario, introduzca la ruta al archivo que incluye las macros definidas por el usuario del proyecto de C++. Asegúrese de que este archivo tiene la siguiente estructura: `#define name value`. Al igual que en el ejemplo anterior, el parámetro `value` es opcional. El valor predeterminado para este parámetro es 1.

Para crear este archivo, abra el proyecto en Microsoft Visual Studio y, a continuación, elija Proyecto, Propiedades, C/C++ y Preprocesador. En Definiciones del preprocesador, elija Editar y copie los nombres y valores en un nuevo archivo de texto. A continuación, para cada cadena del archivo, agregue el siguiente prefijo: `#define` .

4. En Directorios de inclusión externa, introduzca las rutas a las carpetas que incluyen las bibliotecas externas que utiliza en su proyecto de C++.
5. Pulse Aceptar para guardar la configuración del proyecto y cerrar la ventana.

O especifique la configuración de la conversión para cada proyecto de conversión de aplicaciones de C++. Para obtener más información, consulte Convertir el código SQL de la aplicación de C++.

Tras realizar los cambios en el código fuente de la aplicación, cárguelo en el proyecto de AWS SCT.

Para actualizar el código de la aplicación actualizado

1. Expanda el nodo C++ en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea actualizar y abra el menú contextual (clic secundario).
3. Seleccione Actualizar y, a continuación, elija Sí.

   AWS SCT carga el código de la aplicación desde los archivos de origen y elimina los resultados de la conversión. Para conservar los cambios de código realizó en AWS SCT y los resultados de la conversión, cree un nuevo proyecto de conversión de aplicaciones de C++.

   Además, AWS SCT elimina la configuración de la conversión de la aplicación que especificó para la aplicación seleccionada. Tras cargar el código de aplicación actualizado, AWS SCT aplica los valores predeterminados de la configuración del proyecto.

Para eliminar un proyecto de conversión de aplicaciones de C++

1. Expanda el nodo C++ en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea eliminar y abra el menú contextual (clic secundario).
3. Elija Eliminar y a continuación haga clic en Aceptar.

Crear un informe de evaluación de la conversión de la aplicación de C++ en AWS SCT

El Informe de evaluación de la conversión de la aplicación de C++ proporciona información sobre cómo convertir el código SQL incrustado en su aplicación de C++ a un formato compatible con la base de datos de destino. El informe de evaluación proporciona detalles de la conversión para todos los puntos de ejecución de SQL y todos los archivos de código fuente. El informe de evaluación también incluye una lista de elementos de código SQL que AWS SCT no puede convertir.

Para crear un informe de evaluación de la conversión de la aplicación de C++

1. Expanda el nodo C++ en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea convertir y abra el menú contextual (clic secundario).
3. Seleccione Convertir.
4. En el menú Ver, seleccione Ver informe de evaluación.
5. Consulte la pestaña Resumen.

   En la pestaña Resumen que se muestra a continuación aparece la información resumida del informe de evaluación de la aplicación de C++. Muestra los resultados de la conversión de todos los puntos de ejecución de SQL y de todos los archivos de código fuente.
6. Seleccione Guardar instrucciones en JSON para guardar el código SQL extraído de la aplicación de Java como un archivo JSON.
7. (Opcional) Guarde una copia local del informe como archivo PDF o como un archivo de valores separados por comas (CSV):

   - Seleccione Guardar en PDF en la parte superior derecha para guardar el informe como un archivo PDF.

     El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de la aplicación.
   - Seleccione Guardar en CSV en la parte superior derecha para guardar el informe como un archivo CSV.

     El archivo CSV contiene elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir el código SQL.

Conversión del código SQL en aplicaciones de Java con AWS SCT

Para una conversión de Oracle a PostgreSQL, puede utilizar AWS Schema Conversion Tool para convertir el código SQL incrustado en sus aplicaciones de Java. Este conversor de aplicaciones de Java específico comprende la lógica de la aplicación. Recopila las declaraciones que se encuentran en diferentes objetos de la aplicación, como funciones, parámetros, variables locales, etc.

Gracias a este análisis profundo, el conversor de código SQL de aplicaciones de Java proporciona mejores resultados de conversión que el conversor genérico.

Si su aplicación de Java utiliza el marco MyBatis para interactuar con las bases de datos, puede utilizar AWS SCT para convertir instrucciones SQL incrustadas en anotaciones y archivos XML de MyBatis. Para entender la lógica de estas instrucciones SQL, AWS SCT utiliza el archivo de configuración de MyBatis. AWS SCT puede detectar automáticamente este archivo en la carpeta de la aplicación o puede introducir la ruta de acceso a este archivo manualmente.

Crear proyectos de conversión de aplicaciones de Java en AWS SCT

Puede crear un proyecto de conversión de aplicaciones de Java solo para convertir esquemas de bases de datos Oracle en esquemas de bases de datos PostgreSQL. Agregue una regla de asignación a su proyecto que incluya un esquema de Oracle de origen y una base de datos PostgreSQL de destino. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT.

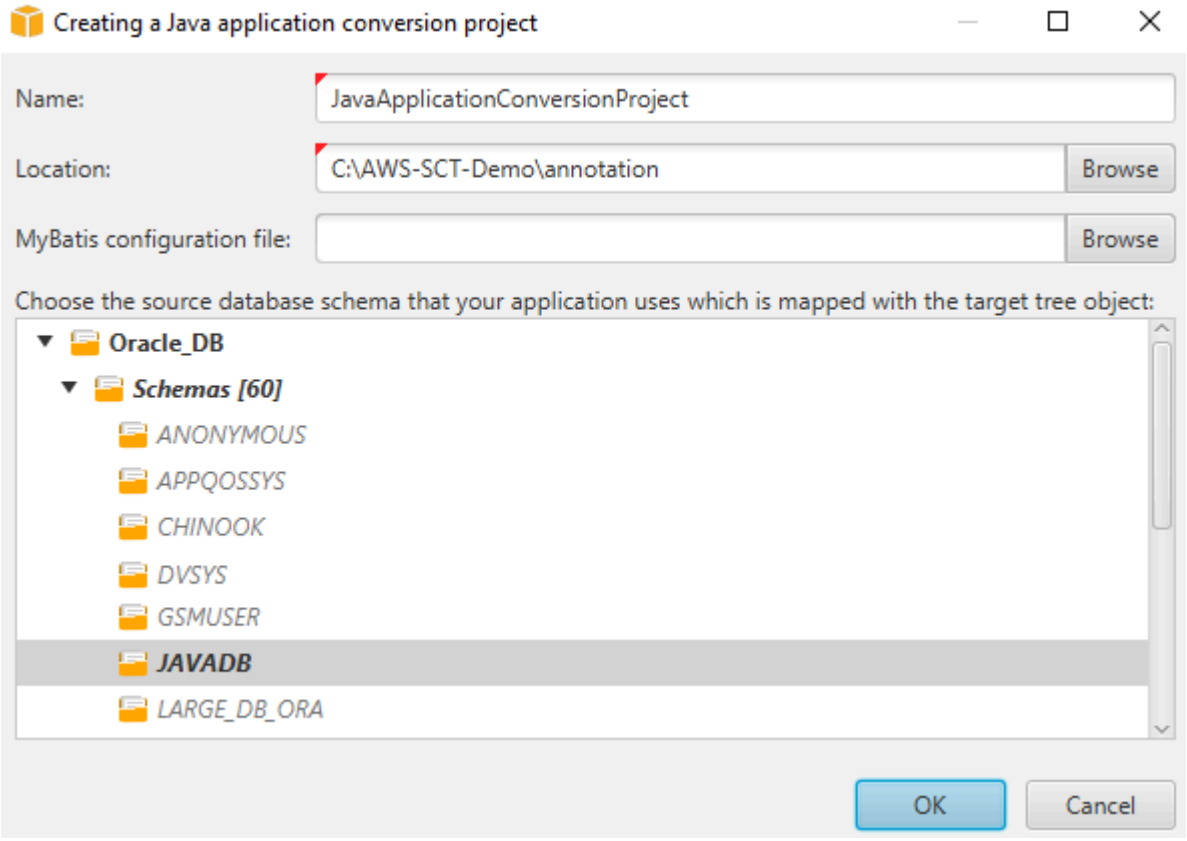
Puede agregar varios proyectos de conversión de aplicaciones en un único AWS SCT proyecto. Utilice el siguiente procedimiento para crear un proyecto de conversión de aplicaciones de Java.

Para crear un proyecto de conversión de aplicaciones de Java

1. Cree un proyecto de conversión de bases de datos y agregue una base de datos Oracle de origen. Para obtener más información, consulte Creación de un proyecto de AWS SCT y Cómo agregar servidores de bases de datos a un proyecto de AWS SCT.
2. Agregue una regla de asignación que incluya la base de datos Oracle de origen y una base de datos PostgreSQL de destino. Puede agregar una base de datos PostgreSQL de destino o utilizar una plataforma de base de datos PostgreSQL de destino virtual en una regla de asignación. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT y Usar destinos virtuales.
3. En el menú Ver, seleccione Vista principal.

4. En el menú Aplicaciones, elija Aplicación de Java nueva.

   Aparecerá el cuadro de diálogo Proyecto de conversión de aplicación de Java nuevo.



5. En Nombre, escriba un nombre para el proyecto de conversión de la aplicación de Java. Puesto que cada proyecto de conversión de esquemas de bases de datos puede tener uno o varios proyectos secundarios de conversión de aplicaciones, elija un nombre significativo si va agrega varios proyectos.
6. En Ubicación, escriba la ubicación del código fuente de su aplicación.
7. (Opcional) En Archivo de configuración de MyBatis introduzca la ruta al archivo de configuración de MyBatis. AWS SCTanaliza la carpeta de la aplicación para detectar este archivo automáticamente. Si este archivo no se encuentra en la carpeta de la aplicación o si utiliza varios archivos de configuración, introduzca la ruta manualmente.
8. En el árbol de origen, elija el esquema utilizado por la aplicación. Asegúrese de que este esquema forme parte de una regla de asignación. AWS SCT resalta en negrita los esquemas que forman parte de una regla de asignación.
9. Seleccione Aceptar para crear su proyecto de conversión de la aplicación de Java.
10. Busque su proyecto de conversión de la aplicación de Java en el nodo Aplicaciones del panel izquierdo.

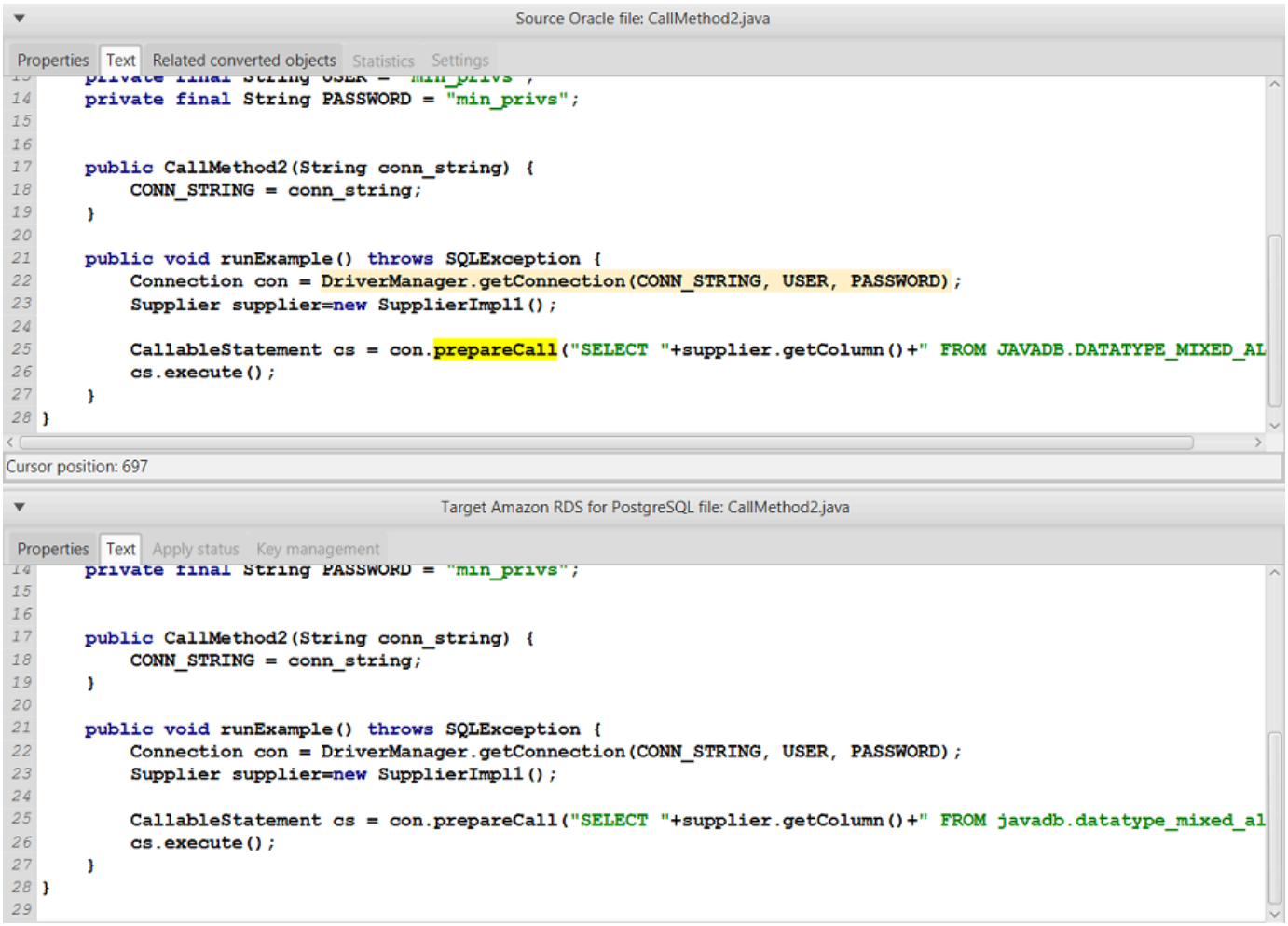
Conversión del código SQL de la aplicación de Java en AWS SCT

Tras agregar la aplicación de Java al proyecto de AWS SCT, convierta el código SQL de esta aplicación a un formato compatible con la plataforma de base de datos de destino. Utilice el siguiente procedimiento para analizar y convertir su código SQL incrustado en su aplicación de Java en AWS Schema Conversion Tool.

Para convertir el código SQL

1. Expanda el nodo Java en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea convertir y abra el menú contextual (clic secundario).
3. Seleccione Convertir. AWS SCT analiza los archivos de código fuente, determina la lógica de la aplicación y carga los metadatos del código en el proyecto. Estos metadatos de código incluyen clases, objetos, métodos, variables globales, interfaces, etc. de Java.

   En el panel de la base de datos de destino, AWS SCT crea una estructura de carpetas similar a la del proyecto de la aplicación de origen. Aquí puede revisar el código de la aplicación convertida.



4. Guarde el código de la aplicación convertida. Para obtener más información, consulte Cómo guardar el código de la aplicación convertida.

Sus aplicaciones de Java pueden incluir código SQL que interactúa con diferentes bases de datos de origen. Puede migrar a PostgreSQL varias de estas bases de datos de origen. En este caso, asegúrese de no convertir el código SQL que interactúa con las bases de datos que excluyó del ámbito de la migración. Puede excluir los archivos de origen de su aplicación de Java del ámbito de la conversión. Para ello, desactive las casillas de verificación de los nombres de los archivos que desee excluir del ámbito de la conversión.

Tras modificar el ámbito de la conversión, AWS SCT seguirá analizando el código SQL de todos los archivos de origen de sus aplicaciones de Java. A continuación, AWS SCT copia en la carpeta de destino todos los archivos de origen que excluyó del ámbito de la conversión. Esta operación permite compilar la aplicación después de guardar los archivos de la aplicación convertidos.

Cómo guardar el código de la aplicación convertida con AWS SCT

Utilice el siguiente procedimiento para guardar el código de la aplicación convertida.

Para guardar el código de la aplicación convertida

1. Expanda el nodo Java en Aplicaciones en el panel de la base de datos de destino.
2. Elija la aplicación convertida y elija Guardar.
3. Introduzca la ruta a la carpeta para guardar el código de la aplicación convertida y seleccione Seleccionar carpeta.

Si la aplicación de Java de origen utiliza el framework MyBatis, actualice el archivo de configuración para que funcione con la nueva base de datos.

Administración de proyectos de conversión de aplicaciones de Java en AWS SCT

Puede agregar varios proyectos de conversión de aplicaciones de Java, actualizar el código de la aplicación del proyecto de AWS SCT o eliminar un proyecto de conversión de aplicaciones de Java del proyecto de AWS SCT.

Para agregar un proyecto de conversión de aplicaciones de Java adicional

1. Expanda el nodo Aplicaciones en el panel izquierdo.
2. Seleccione el nodo Java y abra el menú contextual (clic secundario).
3. Elija Aplicación nueva.
4. Introduzca la información necesaria para crear un proyecto de conversión de aplicaciones de Java nuevo. Para obtener más información, consulte Crear proyectos de conversión de aplicaciones de Java.

Tras realizar los cambios en el código fuente de la aplicación, cárguelo en el proyecto de AWS SCT.

Para actualizar el código de la aplicación actualizado

1. Expanda el nodo Java en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea actualizar y abra el menú contextual (clic secundario).

3. Seleccione Actualizar y, a continuación, elija Sí.

   AWS SCT carga el código de la aplicación desde los archivos de origen y elimina los resultados de la conversión. Para conservar los cambios de código que realizó en AWS SCT y los resultados de la conversión, cree un nuevo proyecto de conversión de aplicaciones de Java.

Si la aplicación de Java de origen utiliza el marco MyBatis, AWS SCT utilizará el archivo de configuración de MyBatis para analizar el código SQL. Después de modificar este archivo, cárguelo en el proyecto de AWS SCT.

Para editar la ruta al archivo de configuración de MyBatis

1. Expanda el nodo Java en Aplicaciones en el panel izquierdo.
2. Elija la aplicación y, a continuación, elija Configuración.
3. Seleccione Examinar y, a continuación, elija el archivo de configuración de MyBatis.
4. Seleccione Aplicar.
5. En el panel izquierdo, elija la aplicación, abra el menú contextual (clic secundario) y elija Actualizar.

Para eliminar un proyecto de conversión de aplicaciones de Java

1. Expanda el nodo Java en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea eliminar y abra el menú contextual (clic secundario).
3. Elija Eliminar y a continuación haga clic en Aceptar.

Creación de un informe de evaluación de la conversión de la aplicación de Java en AWS SCT

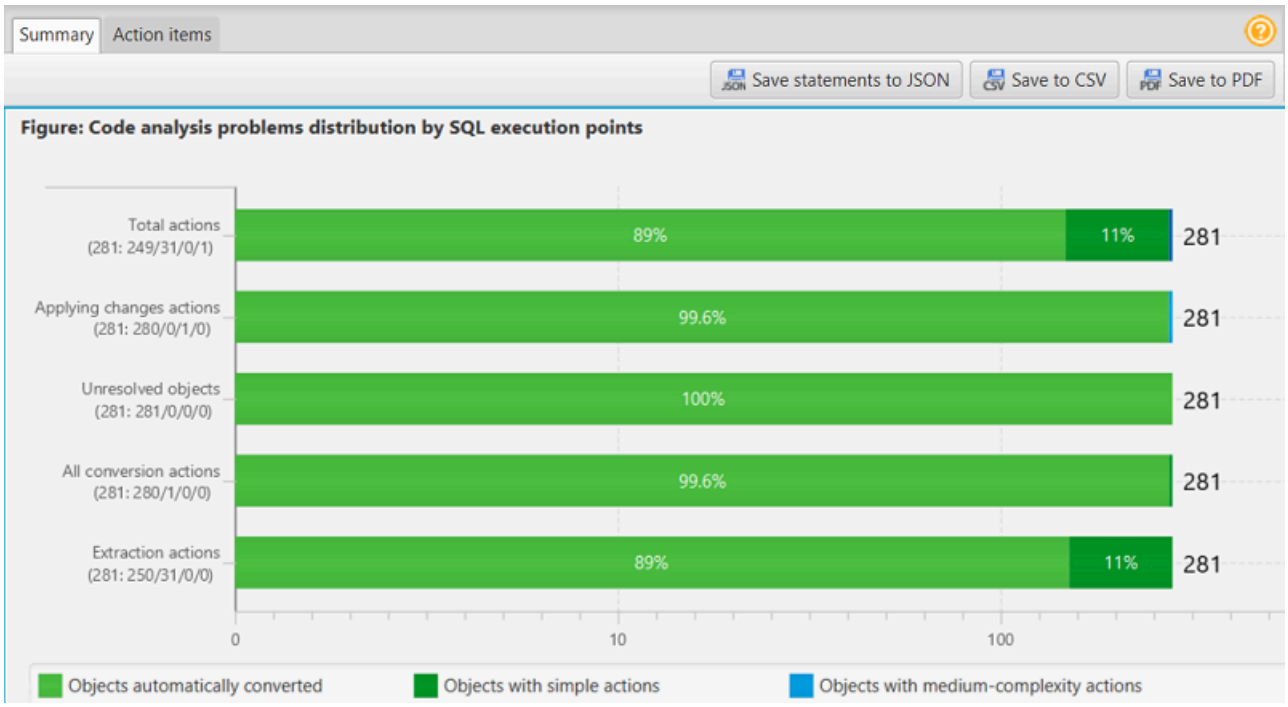
El Informe de evaluación de la conversión de la aplicación de Java proporciona información sobre cómo convertir el código SQL incrustado en su aplicación de Java a un formato compatible con la base de datos de destino. El informe de evaluación proporciona detalles de la conversión para todos los puntos de ejecución de SQL y todos los archivos de código fuente. El informe de evaluación también incluye una lista de elementos de código SQL que AWS SCT no puede convertir.

Utilice el siguiente procedimiento para crear un informe de evaluación de la conversión de la aplicación de Java.

Para crear un informe de evaluación de la conversión de la aplicación de Java

1. Expanda el nodo Java en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea convertir y abra el menú contextual (clic secundario).
3. Seleccione Convertir.
4. En el menú Ver, seleccione Ver informe de evaluación.
5. Consulte la pestaña Resumen.

   En la pestaña Resumen que se muestra a continuación aparece la información del resumen ejecutivo del informe de evaluación de la aplicación de Java. Muestra los resultados de la conversión de todos los puntos de ejecución de SQL y de todos los archivos de código fuente.



6. Seleccione Guardar instrucciones en JSON para guardar el código SQL extraído de la aplicación de Java como un archivo JSON.
7. (Opcional) Guarde una copia local del informe como archivo PDF o como un archivo de valores separados por comas (CSV):

   - Seleccione Guardar en PDF en la parte superior derecha para guardar el informe como un archivo PDF.

     El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de la aplicación.

- Seleccione Guardar en CSV en la parte superior derecha para guardar el informe como un archivo CSV.

  El archivo CSV contiene elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir el código SQL.

Conversión del código SQL en aplicaciones de Pro*C con AWS SCT

Para una conversión de Oracle a PostgreSQL, puede utilizar AWS Schema Conversion Tool (AWS SCT) para convertir el código SQL incrustado en sus aplicaciones de Pro*C. Este conversor de aplicaciones de Pro*C específico comprende la lógica de la aplicación. Recopila las declaraciones que se encuentran en diferentes objetos de la aplicación, como funciones, parámetros, variables locales, etc.

Gracias a este análisis profundo, el conversor de código SQL de aplicaciones de Pro*C proporciona mejores resultados de conversión que el conversor genérico.

Creación de proyectos de conversión de aplicaciones de Pro*C en AWS SCT

Puede crear un proyecto de conversión de aplicaciones de Pro*C solo para convertir esquemas de bases de datos Oracle en esquemas de bases de datos PostgreSQL. Agregue una regla de asignación a su proyecto que incluya un esquema de Oracle de origen y una base de datos PostgreSQL de destino. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT.

Puede agregar varios proyectos de conversión de aplicaciones en un único proyecto de AWS SCT. Utilice el siguiente procedimiento para crear un proyecto de conversión de aplicaciones de Pro*C.

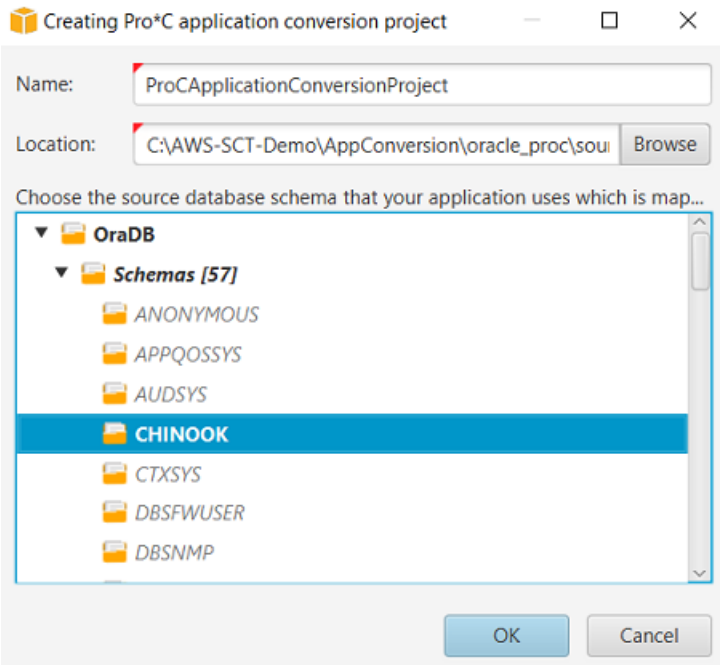
Para crear un proyecto de conversión de aplicaciones de Pro*C

1. Cree un proyecto de conversión de bases de datos y agregue una base de datos Oracle de origen. Para obtener más información, consulte Creación de un proyecto de AWS SCT y Cómo agregar servidores de bases de datos a un proyecto de AWS SCT.
2. Agregue una regla de asignación que incluya la base de datos Oracle de origen y una base de datos PostgreSQL de destino. Puede agregar una base de datos PostgreSQL de destino

o utilizar una plataforma de base de datos PostgreSQL de destino virtual en una regla de asignación. Para obtener más información, consulte Creación de reglas de asignación en AWS SCT y Usar destinos virtuales.

3. En el menú Ver, seleccione Vista principal.
4. En el menú Aplicaciones, elija Aplicación de Pro*C nueva.

   Aparecerá el cuadro de diálogo Proyecto de conversión de aplicación de Pro*C nuevo.



5. En Nombre, escriba un nombre para el proyecto de conversión de la aplicación de Pro*C. Puesto que cada proyecto de conversión de esquemas de bases de datos puede tener uno o varios proyectos secundarios de conversión de aplicaciones, elija un nombre significativo si va agrega varios proyectos.
6. En Ubicación, escriba la ubicación del código fuente de su aplicación.
7. En el árbol de origen, elija el esquema utilizado por la aplicación. Asegúrese de que este esquema forme parte de una regla de asignación. AWS SCT resalta en negrita los esquemas que forman parte de una regla de asignación.
8. Seleccione Aceptar para crear su proyecto de conversión de la aplicación de Pro*C.
9. Busque su proyecto de conversión de la aplicación de Pro*C en el nodo Aplicaciones del panel izquierdo.

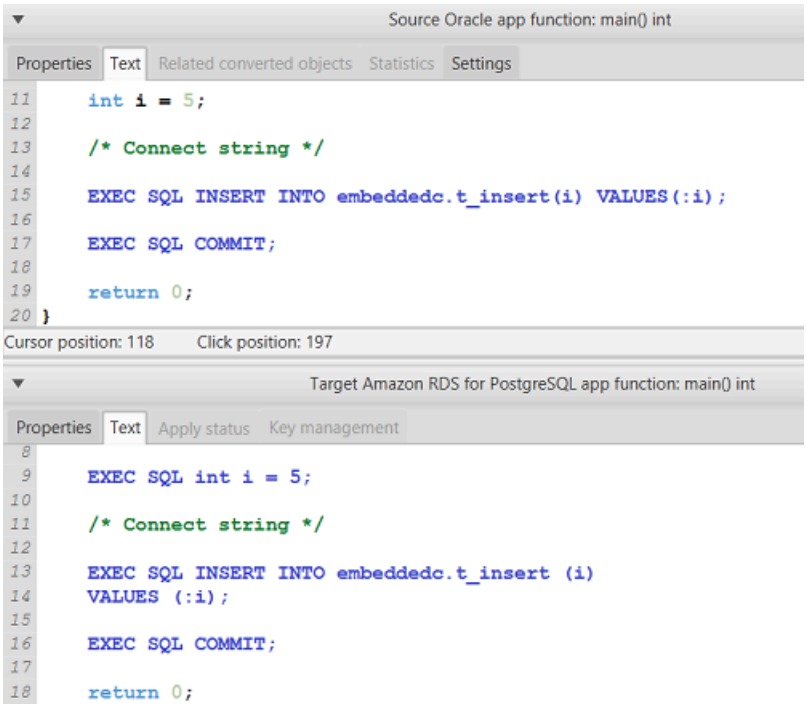
Conversión del código SQL de la aplicación de Pro*C en AWS SCT

Tras agregar la aplicación de Pro*C al proyecto de AWS SCT, convierta el código SQL de esta aplicación a un formato compatible con la plataforma de base de datos de destino. Utilice el siguiente procedimiento para analizar y convertir su código SQL incrustado en su aplicación de Pro*C en AWS Schema Conversion Tool.

Para convertir el código SQL

1. Expanda el nodo Pro*C en Aplicaciones en el panel izquierdo.
2. Elija la aplicación de que desea convertir y, seguidamente, seleccione Configuración.
    a. En Ruta del archivo de encabezado global, introduzca la ruta de acceso a los archivos de encabezado que utiliza el proyecto de la aplicación.
    b. Seleccione Interpretar todas las variables de host no resueltas como para ver todas las variables no resueltas en el código convertido.
    c. Seleccione Utilizar la función de conversión de cadenas de ancho fijo del paquete de extensión para utilizar las funciones del paquete de extensión en el código SQL convertido. AWS SCT incluye los archivos del paquete de extensión en el proyecto de la aplicación.
    d. Seleccione Transformar bloques PL/SQL anónimos en llamadas SQL independientes o funciones almacenadas para crear procedimientos almacenados en la base de datos de destino para todos los bloques PL/SQL anónimos. A continuación, AWS SCT incluye las ejecuciones de estos procedimientos almacenados en el código de la aplicación convertida.
    e. Seleccione Utilizar un flujo de cursores personalizado para mejorar la conversión de los cursores de las bases de datos Oracle.
3. En el panel izquierdo, seleccione la aplicación que desea convertir y abra el menú contextual (clic secundario).
4. Seleccione Convertir. AWS SCT analiza los archivos de código fuente, determina la lógica de la aplicación y carga los metadatos del código en el proyecto. Estos metadatos de código incluyen clases, objetos, métodos, variables globales, interfaces, etc. de Pro*C.

    En el panel de la base de datos de destino, AWS SCT crea una estructura de carpetas similar a la del proyecto de la aplicación de origen. Aquí puede revisar el código de la aplicación convertida.



5. Guarde el código de la aplicación convertida. Para obtener más información, consulte Editar y guardar el código de la aplicación convertida.

Editar y guardar el código de la aplicación convertida con AWS SCT

Puede editar las instrucciones SQL convertidas y utilizar AWS SCT para incrustar este código editado en el código de la aplicación Pro*C convertida. Utilice el siguiente procedimiento para editar el código SQL convertido.

Para editar el código SQL convertido

1. Expanda el nodo Pro*C en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea convertir, abra el menú contextual (clic secundario) y seleccione Convertir.
3. En el menú Ver, seleccione Ver informe de evaluación.
4. Seleccione Guardar instrucciones en CSV para guardar el código SQL extraído de la aplicación de Pro*C como un archivo CSV.
5. Introduzca el nombre del archivo CSV para guardar el código SQL extraído y seleccione Guardar.
6. Edite el código SQL extraído.

7. En el menú Ver, seleccione Vista principal.
8. Expanda el nodo Pro*C en Aplicaciones en el panel de la base de datos de destino.
9. Elija la aplicación convertida, abra el menú contextual (clic secundario) y elija Importar instrucciones de CSV.
10. Seleccione Sí, seleccione el archivo con el código SQL editado y, seguidamente, selecciona Abrir.

AWS SCT divide las instrucciones SQL convertidas en partes y las coloca en los objetos correspondientes del código fuente de la aplicación. Utilice el siguiente procedimiento para guardar el código de la aplicación convertida.

Para guardar el código de la aplicación convertida

1. Expanda el nodo Pro*C en Aplicaciones en el panel de la base de datos de destino.
2. Elija la aplicación convertida y elija Guardar.
3. Introduzca la ruta a la carpeta para guardar el código de la aplicación convertida y seleccione Seleccionar carpeta.

Administración de proyectos de conversión de aplicaciones de Pro*C en AWS SCT

Puede agregar varios proyectos de conversión de aplicaciones de Pro*C, actualizar el código de la aplicación del proyecto de AWS SCT o eliminar un proyecto de conversión de aplicaciones de Pro*C del proyecto de AWS SCT.

Para agregar un proyecto de conversión de aplicaciones de Pro*C adicional

1. Expanda el nodo Aplicaciones en el panel izquierdo.
2. Seleccione el nodo Pro*C y abra el menú contextual (clic secundario).
3. Elija Aplicación nueva.
4. Introduzca la información necesaria para crear un proyecto de conversión de aplicaciones de Pro*C nuevo. Para obtener más información, consulte Creación de proyectos de conversión de aplicaciones de Pro*C.

Tras realizar los cambios en el código fuente de la aplicación, cárguelo en el proyecto de AWS SCT.

Para actualizar el código de la aplicación actualizado

1. Expanda el nodo Pro*C en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea actualizar y abra el menú contextual (clic secundario).
3. Seleccione Actualizar y, a continuación, elija Sí.

   AWS SCT carga el código de la aplicación desde los archivos de origen y elimina los resultados de la conversión. Para conservar los cambios de código que realizó en AWS SCT y los resultados de la conversión, cree un nuevo proyecto de conversión de aplicaciones de Pro*C.

Para eliminar un proyecto de conversión de aplicaciones de Pro*C

1. Expanda el nodo Pro*C en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea eliminar y abra el menú contextual (clic secundario).
3. Elija Eliminar y a continuación haga clic en Aceptar.

Creación de un informe de evaluación de la conversión de la aplicación de Pro*C en AWS SCT

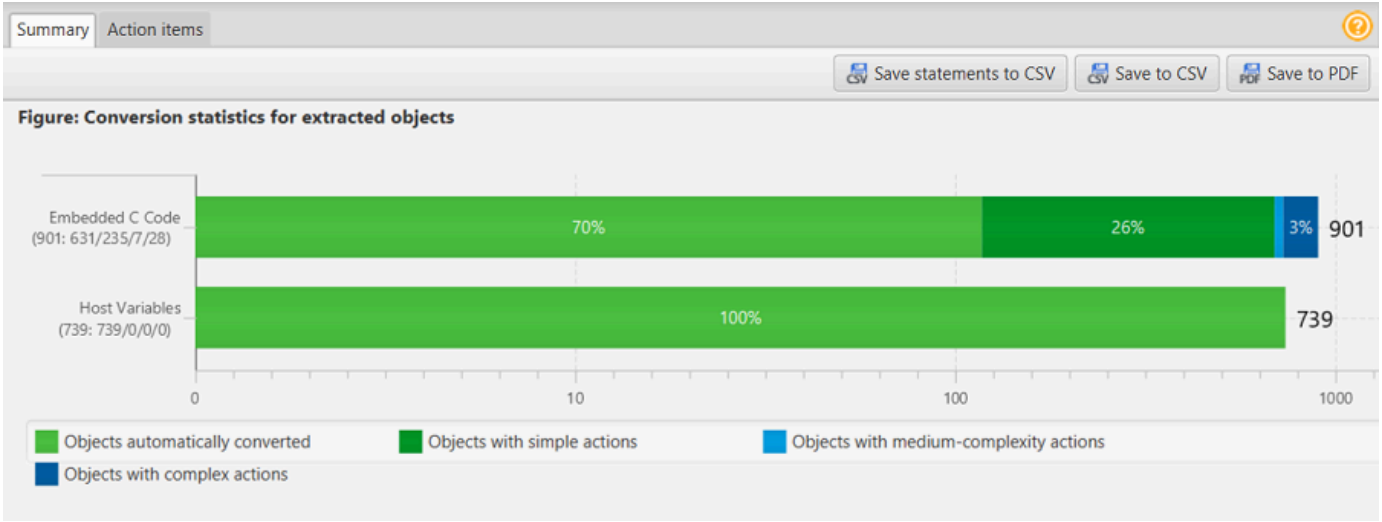
El Informe de evaluación de la conversión de la aplicación de Pro*C proporciona información sobre cómo convertir el código SQL incrustado en su aplicación de Pro*C a un formato compatible con la base de datos de destino. El informe de evaluación proporciona detalles de la conversión para todos los puntos de ejecución de SQL y todos los archivos de código fuente. El informe de evaluación también incluye una lista de elementos de código SQL que AWS SCT no puede convertir.

Utilice el siguiente procedimiento para crear un informe de evaluación de la aplicación de Pro*C.

Para crear un informe de evaluación de la conversión de la aplicación de Pro*C

1. Expanda el nodo Pro*C en Aplicaciones en el panel izquierdo.
2. Elija la aplicación que desea convertir y abra el menú contextual (clic secundario).
3. Seleccione Convertir.
4. En el menú Ver, seleccione Ver informe de evaluación.
5. Consulte la pestaña Resumen.

En la pestaña Resumen que se muestra a continuación aparece la información resumida del informe de evaluación de la aplicación de Pro*C. Muestra los resultados de la conversión de todos los puntos de ejecución de SQL y de todos los archivos de código fuente.



6. Seleccione Guardar instrucciones en CSV para guardar el código SQL extraído de la aplicación de Pro*C como un archivo de valores separados por comas (CSV).
7. (Opcional) Guarde una copia local del informe como archivo PDF o como un archivo de valores separados por comas (CSV):

   - Seleccione Guardar en PDF en la parte superior derecha para guardar el informe como un archivo PDF.

     El archivo PDF contiene el resumen ejecutivo, los elementos de acción y las recomendaciones para la conversión de la aplicación.
   - Seleccione Guardar en CSV en la parte superior derecha para guardar el informe como un archivo CSV.

     El archivo CSV contiene elementos de acción, acciones recomendadas y una estimación de la complejidad del esfuerzo manual necesario para convertir el código SQL.

Uso de paquetes AWS SCT de extensión

Un paquete de extensión de AWS SCT es un módulo complementario que simula funciones presentes en una base de datos de origen que son necesarias a la hora de convertir objetos a la base de datos de destino. Antes de poder instalar un paquete de AWS SCT extensión, debe convertir el esquema de la base de datos.

Cada paquete de AWS SCT extensión incluye los siguientes componentes:

- Esquema de base de datos: incluye funciones, procedimientos y tablas de SQL para simular ciertos objetos de bases de datos de procesamiento de transacciones en línea (OLTP) y procesamiento analítico en línea (OLAP), como las secuencias. Además, emula de forma no compatible con la base built-in-functions de datos de origen. El nombre de este esquema tiene el siguiente formato: `aws_`*`database_engine_name`*`_ext`.
- AWS Lambda funciones (para determinadas bases de datos OLTP): incluye AWS Lambda funciones que emulan funciones complejas de bases de datos, como la programación de tareas y el envío de correos electrónicos.
- Bibliotecas personalizadas para bases de datos OLAP: incluye un conjunto de bibliotecas de Java y Python que puede usar para migrar scripts de Microsoft SQL Server Integration Services (SSIS) para extraer, transformar y cargar (ETL) a AWS Glue o. AWS Glue Studio

  Las bibliotecas de Java incluyen los siguientes módulos:

  - `spark-excel_2.11-0.13.1.jar`: para emular la funcionalidad de los componentes de origen y destino de Excel.
  - `spark-xml_2.11-0.9.0.jar`, `poi-ooxml-schemas-4.1.2.jar` y `xmlbeans-3.1.0.jar`: para simular la funcionalidad del componente de origen de XML.

  Las bibliotecas de Python incluyen los siguientes módulos:

  - `sct_utils.py`: para simular los tipos de datos de origen y preparar los parámetros para la consulta SQL de Spark.
  - `ssis_datetime.py`: para simular las funciones integradas de fecha y hora.
  - `ssis_null.py`: para simular las funciones integradas de `ISNULL` y `REPLACENULL`.
  - `ssis_string.py`: para simular funciones integradas de cadenas.

  Para obtener más información sobre estas bibliotecas, consulte Uso de bibliotecas personalizadas para los paquetes de extensión de AWS SCT.

Los paquetes de AWS SCT extensión se pueden aplicar de dos maneras:

- AWS SCT puede aplicar automáticamente un paquete de extensiones al aplicar un script de base de datos de destino seleccionando Aplicar a la base de datos en el menú contextual. AWS SCT aplica el paquete de extensión antes de aplicar todos los demás objetos del esquema.
- Para aplicar manualmente un paquete de extensión, seleccione la base de datos de destino y, a continuación, elija Aplicar paquete de extensión para desde el menú contextual (clic secundario). La aplicación automática es suficiente para la mayoría de las situaciones. Sin embargo, es posible que desee aplicar el paquete de forma manual si se elimina accidentalmente.

Cada vez que se aplica un paquete de AWS SCT extensión a un banco de datos de destino, los componentes se sobrescriben y se AWS SCT muestra una notificación al respecto. Para desactivar estas notificaciones, seleccione Configuración, Configuración global, Notificaciones y, a continuación, seleccione Ocultar la alerta de sustitución del paquete de extensión.

Para una conversión de Microsoft SQL Server a PostgreSQL, puede utilizar el paquete de extensión de SQL Server a PostgreSQL en AWS SCT. Este paquete de extensión simula SQL Server Agent y SQL Server Database Mail. Para más información, consulte Simular el Agente SQL Server en PostgreSQL con un paquete de extensión y Simular el correo electrónico de base de datos de SQL Server en PostgreSQL con un paquete de extensión.

A continuación, encontrará más información sobre cómo trabajar con paquetes de AWS SCT extensiones.

Temas

- Permisos para usar el paquete AWS SCT de extensiones
- Uso del esquema del paquete de extensión
- Uso de bibliotecas personalizadas para los paquetes de extensión de AWS SCT
- Uso de las AWS Lambda funciones del paquete de AWS SCT extensión
- Configuración de las funciones del paquete de AWS SCT extensión

Permisos para usar el paquete AWS SCT de extensiones

El paquete de AWS SCT extensión para Amazon Aurora emula el envío de correo, la programación de tareas, la creación de colas y otras operaciones mediante funciones. AWS Lambda Al aplicar el paquete de AWS SCT extensión a la base de datos Aurora de destino, AWS SCT crea una nueva

función AWS Identity and Access Management (IAM) y una política de IAM integrada. A continuación, AWS SCT crea una nueva función Lambda y configura el clúster de base de datos Aurora para las conexiones salientes a. AWS Lambda Para ejecutar estas operaciones, debe conceder los siguientes permisos necesarios al usuario de IAM:

- `iam:CreateRole`— para crear un nuevo rol de IAM para su cuenta. AWS
- `iam:CreatePolicy`— para crear una nueva política de IAM para su AWS cuenta.
- `iam:AttachRolePolicy`: para adjuntar la política especificada al rol de IAM.
- `iam:PutRolePolicy`: para actualizar un documento de política insertado que está integrado en el rol de IAM especificado.
- `iam:PassRole`: para transferir el rol de IAM especificado al motor de reglas.
- `iam:TagRole`: para agregar etiquetas a un rol de IAM.
- `iam:TagPolicy`: para agregar etiquetas a una política de IAM.
- `lambda:ListFunctions`: para ver la lista de sus funciones de Lambda.
- `lambda:ListTags`: para ver la lista de etiquetas de sus funciones de Lambda.
- `lambda:CreateFunction`: para crear una nueva función de Lambda.
- `rds:AddRoleToDBCluster`: para asociar un rol de IAM a su clúster de base de datos de Aurora.

El paquete de AWS SCT extensiones para Amazon Redshift emula las funciones de base del almacén de datos de origen que se requieren al aplicar objetos convertidos a Amazon Redshift. Antes de aplicar el código convertido a Amazon Redshift, debe aplicar el paquete de extensión para Amazon Redshift. Para ello, incluya la acción `iam:SimulatePrincipalPolicy` en la política de IAM.

AWS SCT utiliza el simulador de políticas de IAM para comprobar los permisos necesarios para instalar el paquete de extensión Amazon Redshift. El simulador de política de IAM puede mostrar un mensaje de error, aunque el usuario de IAM se haya configurado correctamente. Se trata de un problema conocido del simulador de política de IAM. Además, este simulador muestra un mensaje de error cuando la política de IAM no incluye esa acción de `iam:SimulatePrincipalPolicy`. En estos casos, puede ignorar el mensaje de error y aplicar el paquete de extensión utilizando el asistente del paquete de extensión. Para obtener más información, consulte Aplicación del paquete de extensión.

Uso del esquema del paquete de extensión

Al convertir su esquema de almacenamiento de datos o base de datos, AWS SCT añade un esquema adicional a su base de datos de destino. Este esquema implementa las funciones del sistema SQL de la base de datos de origen que son necesarias al escribir su esquema convertido en la base de datos de destino. El esquema adicional se denomina esquema del paquete de extensión.

El esquema del paquete de extensión para bases de datos OLTP se nombra en función a la base de datos de origen, de la siguiente manera:

- Microsoft SQL Server: `AWS_SQLSERVER_EXT`
- MySQL: `AWS_MYSQL_EXT`
- Oracle: `AWS_ORACLE_EXT`
- PostgreSQL: `AWS_POSTGRESQL_EXT`

El esquema del paquete de extensión para aplicaciones data warehouse OLAP se nombra en función del almacén de base de datos de origen, de la siguiente manera:

- Greenplum: `AWS_GREENPLUM_EXT`
- Microsoft SQL Server: `AWS_SQLSERVER_EXT`
- Netezza: `AWS_NETEZZA_EXT`
- Oracle: `AWS_ORACLE_EXT`
- Teradata: `AWS_TERADATA_EXT`
- Vertica: `AWS_VERTICA_EXT`

Uso de bibliotecas personalizadas para los paquetes de extensión de AWS SCT

En algunos casos, no AWS SCT puede convertir las características de la base de datos de origen en entidades equivalentes en la base de datos de destino. El paquete de AWS SCT extensión correspondiente contiene bibliotecas personalizadas que emulan algunas funciones de la base de datos de origen en la base de datos de destino.

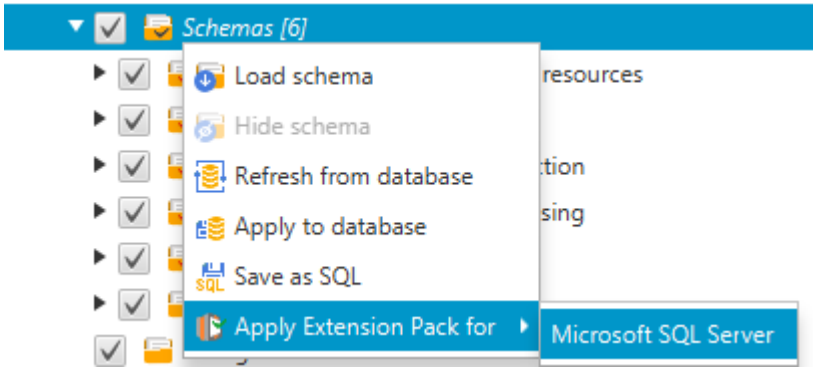
Si va a convertir una base de datos transaccional, consulte Uso de las AWS Lambda funciones del paquete de AWS SCT extensión .

Aplicación del paquete de extensión

Puede aplicar el paquete de AWS SCT extensión mediante el asistente del paquete de extensiones o al aplicar el código convertido a la base de datos de destino.

Para aplicar el paquete de extensión mediante el asistente del paquete de extensión

1. En el AWS Schema Conversion Toolárbol de la base de datos de destino, abra el menú contextual (haga clic con el botón derecho), seleccione Aplicar paquete de extensión para y, a continuación, elija la plataforma de base de datos de origen.



   Aparecerá el asistente de paquete de extensión.
2. Lea la página Welcome y seleccione Next.
3. En la página Configuración del perfil de AWS , haga lo siguiente:

   - Si está volviendo a instalar solamente el esquema del paquete de extensión, seleccione Skip this step for now y, a continuación, Next. La opción Omitir este paso por ahora solo está disponible para las bases de datos de procesamiento de transacciones en línea (OLTP).
   - Si está cargando una biblioteca nueva, facilite las credenciales para conectarse a su Cuenta de AWS. Utilice este paso únicamente cuando convierta bases de datos OLAP o scripts de ETL. Puede usar sus credenciales AWS Command Line Interface (AWS CLI) si las tiene AWS CLI instaladas. También puede utilizar las credenciales que ya haya almacenado en un perfil en la configuración global de la aplicación y tenga asociadas con el proyecto. Si es necesario, elija Navegar a la configuración global para configurar o asociar un perfil diferente a su AWS SCT proyecto. Para obtener más información, consulte Almacenamiento de perfiles de servicio de AWS en AWS SCT.
4. Si va a cargar una biblioteca nueva, seleccione Necesito cargar una biblioteca en la página Carga de bibliotecas. Utilice este paso únicamente cuando convierta bases de datos OLAP o

scripts de ETL. A continuación, proporcione la ruta de Amazon S3 y, luego, seleccione Cargar biblioteca a S3.

Si ya ha cargado la biblioteca, seleccione la opción Ya tengo bibliotecas cargadas, usar mi bucket de S3 existente en la página Carga de bibliotecas. A continuación, proporcione la ruta de Amazon S3.

Cuando haya terminado, elija Next.

5. En la página Simulación de funciones, seleccione Crear paquete de extensión. Aparecerán mensajes con el estado de las operaciones del paquete de extensión.

   Cuando haya terminado, seleccione Finish.

Para aplicar el paquete de extensión al aplicar el código convertido

1. Especifique el bucket de Amazon S3 en su perfil de AWS servicio. Utilice este paso únicamente cuando convierta bases de datos OLAP o scripts de ETL. Para obtener más información, consulte Almacenamiento de perfiles de servicio de AWS en AWS SCT.

   Compruebe que el bucket de Amazon S3 incluye los siguientes permisos:

```
{
    "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"]
      }
    ]
}
```

En el ejemplo anterior, sustituya *111122223333:user/* por el nombre de su usuario DataExtractionAgentName de IAM.

2. Convierta sus esquemas de almacenamiento de datos de origen. Para obtener más información, consulte Convertir esquemas de almacenamiento de datos a Amazon Redshift.
3. En el panel derecho, elija el esquema convertido.
4. Abra el menú contextual (clic con el botón secundario) del elemento del esquema y seleccione Aplicar a base de datos.
5. AWS SCT genera paquetes de extensión con los componentes necesarios y agrega el esquema en el `aws_`*database_engine_name*`_ext` árbol de destino. A continuación, AWS SCT aplica el código convertido y el esquema del paquete de extensiones al almacén de datos de destino.

   Si utiliza una combinación de Amazon Redshift y AWS Glue como plataforma de base de datos de destino, AWS SCT añade un esquema adicional al paquete de extensiones.

Uso de las AWS Lambda funciones del paquete de AWS SCT extensión

AWS SCT proporciona un paquete de extensiones que contiene funciones Lambda para correo electrónico, programación de tareas y otras características para bases de datos alojadas en Amazon EC2.

Uso de AWS Lambda funciones para emular la funcionalidad de la base de datos

En algunos casos, las características de la base de datos de origen no se pueden convertir a características de Amazon RDS equivalentes. Algunos ejemplos son las llamadas de envío de correos electrónicos de Oracle que utilizan `UTL_SMTP` y las tareas de Microsoft SQL Server que utilizan un programador de trabajos. Si aloja y autogestiona una base de datos en Amazon EC2, puede emular estas características AWS sustituyéndolas por servicios.

El asistente de paquetes de AWS SCT extensión le ayuda a instalar, crear y configurar funciones de Lambda para emular el correo electrónico, la programación de tareas y otras funciones.

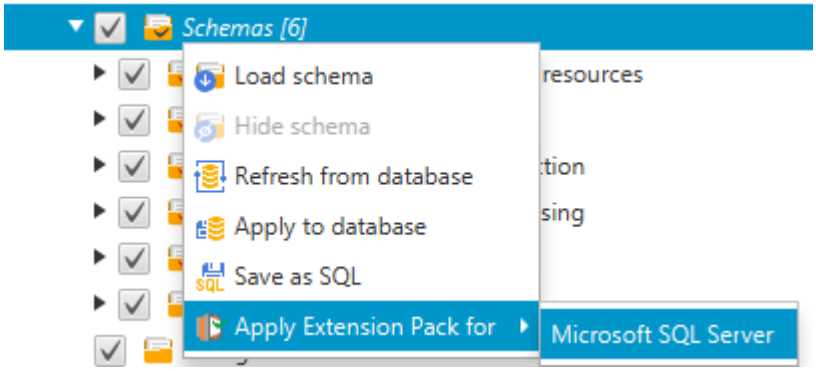
Aplicación del paquete de extensión para admitir funciones de Lambda

Puede aplicar el paquete de extensión para admitir funciones de Lambda mediante el asistente del paquete de extensión o al aplicar el código convertido a la base de datos de destino.

> **Important**
>
> Las funciones AWS de emulación de servicios solo son compatibles con las bases de datos instaladas y autogestionadas en Amazon EC2. No instale las características de simulación de servicios si la base de datos de destino está en una instancia de base de datos de Amazon RDS.

Para aplicar el paquete de extensión mediante el asistente del paquete de extensión

1. En el AWS Schema Conversion Toolárbol de la base de datos de destino, abra el menú contextual (haga clic con el botón derecho), seleccione Aplicar paquete de extensión para y, a continuación, elija la plataforma de base de datos de origen.



   Aparecerá el asistente de paquete de extensión.
2. Lea la página Welcome y seleccione Next.
3. En la página Configuración del perfil de AWS , haga lo siguiente:

   - Si está volviendo a instalar solamente el esquema del paquete de extensión, seleccione Skip this step for now y, a continuación, Next.
   - Si va a instalar AWS servicios, proporcione las credenciales para conectarse a su Cuenta de AWS. Puede usar sus AWS CLI credenciales si las tiene AWS CLI instaladas. También puede

utilizar las credenciales que ya haya almacenado en un perfil en la configuración global de la aplicación y tenga asociadas con el proyecto. Si fuera necesario, seleccione Navigate to Project Settings para asociar al proyecto un perfil diferente. Si fuera necesario, seleccione Configuración global para crear un nuevo perfil. Para obtener más información, consulte Almacenamiento de perfiles de servicio de AWS en AWS SCT.

4. En la página Email Sending Service, haga lo siguiente:

   - Si está volviendo a instalar solamente el esquema del paquete de extensión, seleccione Skip this step for now y, a continuación, Next.
   - Si está instalando AWS servicios y tiene una función Lambda existente, puede proporcionarla. De lo contrario, el asistente la creará por usted. Cuando haya terminado, elija Next.

5. En la página Job Emulation Service, haga lo siguiente:

   - Si está volviendo a instalar solamente el esquema del paquete de extensión, seleccione Skip this step for now y, a continuación, Next.
   - Si está instalando AWS servicios y tiene una función Lambda existente, puede proporcionarla. De lo contrario, el asistente la creará por usted. Cuando haya terminado, elija Next.

6. En la página Simulación de funciones, seleccione Crear paquete de extensión. Aparecerán mensajes con el estado de las operaciones del paquete de extensión.

   Cuando haya terminado, seleccione Finish.

> **Note**
>
> Para actualizar un paquete de extensión y sobrescribir los componentes del paquete de extensión anterior, asegúrese de utilizar la última versión de. AWS SCT Para obtener más información, consulte Instalación, verificación y actualización AWS SCT.

Configuración de las funciones del paquete de AWS SCT extensión

El paquete de extensión contiene funciones que debe configurar antes de su uso. La constante `CONVERSION_LANG` define el idioma que utiliza el service pack. Las funciones están disponibles en inglés y alemán.

Para configurar el idioma en inglés o alemán, realice el siguiente cambio en el código de la función. Busque la siguiente declaración constante:

```
CONVERSION_LANG CONSTANT VARCHAR := '';
```

`CONVERSION_LANG`Para ponerlo en inglés, cambie la línea por la siguiente:

```
CONVERSION_LANG CONSTANT VARCHAR := 'English';
```

Para `CONVERSION_LANG` configurarla en inglés, cambie la línea por la siguiente:

```
CONVERSION_LANG CONSTANT VARCHAR := 'Deutsch';
```

Establezca este ajuste para las siguientes funciones:

- `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`

Prácticas recomendadas para AWS SCT

Obtenga información sobre prácticas recomendadas y opciones al usar la AWS Schema Conversion Tool (AWS SCT).

Configurar memoria adicional

Para convertir grandes esquemas de bases de datos, como una base de datos con 3 500 procedimientos almacenados, puede configurar la cantidad de memoria disponible para AWS Schema Conversion Tool.

Para modificar la cantidad de memoria que consume AWS SCT

1. En el menú Configuración, seleccione Configuración global y, a continuación, seleccione Opciones de JVM.
2. Elija Editar archivo de configuración y elija el editor de texto para abrirlo.
3. Edite la sección `JavaOptions` para definir la memoria disponible mínima y máxima. El siguiente ejemplo establece el mínimo en 4 GB y el máximo en 40 GB.

   ```
   [JavaOptions]
   -Xmx40960M
   -Xms4096M
   ```

   Recomendamos establecer la memoria mínima disponible en al menos 4 GB.
4. Guarde el archivo de configuración, seleccione Aceptar y reinicie AWS SCT para aplicar los cambios.

Configuración de la carpeta de proyecto predeterminada

AWS SCT utiliza la carpeta de proyecto para almacenar los archivos del proyecto, guardar los informes de evaluación y almacenar el código convertido. De forma predeterminada, AWS SCT almacena todos los archivos en la carpeta de la aplicación. Puede especificar otra carpeta como carpeta de proyecto predeterminada.

Para cambiar la carpeta de proyecto predeterminada

1. En el menú Configuración, seleccione Configuración global y, a continuación, seleccione Ruta de archivo.
2. En Ruta de archivo de proyecto predeterminada, introduzca la ruta a la carpeta de proyecto predeterminada.
3. Elija Aplicar y, después, Aceptar.

Incrementación de la velocidad de migración de los datos

Para migrar conjuntos de datos de gran tamaño, como un conjunto de tablas con más de 1 TB de datos, puede que desee aumentar la velocidad de migración. Cuando utiliza agentes de extracción de datos, la velocidad de las migraciones de datos depende de varios factores. Estos factores incluyen el número de segmentos del clúster de Amazon Redshift de destino, el tamaño de un archivo fragmentado en la tarea de migración, la RAM disponible en el PC en el que ejecuta los agentes de extracción de datos, etc.

Para aumentar la velocidad de migración de los datos, le recomendamos que ejecute varias sesiones de migración de prueba con pequeños conjuntos de datos de producción. Además, le recomendamos que ejecute los agentes de extracción de datos en un PC con un SSD de al menos 500 GB de tamaño. Durante estas sesiones de prueba, modifique los distintos parámetros de migración y supervise la utilización del disco para averiguar la configuración que garantiza la máxima velocidad de migración de datos. A continuación, utilice esta configuración para migrar todo el conjunto de datos.

Cómo aumentar la información de registro

Puede aumentar la información de registro producida por AWS SCT al convertir sus bases de datos, scripts y SQL de aplicaciones. Si bien el aumento de la información de registro puede ralentizar la conversión, los cambios pueden ayudarle a proporcionar información sólida a AWS Support en caso de que surjan errores.

AWS SCT almacena los registros en el entorno local. Puede ver estos archivos de registro y compartirlos con Soporte de AWS o con los desarrolladores de AWS SCT para solucionar problemas.

Para cambiar la configuración de registro

1. En el menú Configuración, seleccione Configuración global y, a continuación, seleccione Registro.
2. En Ruta de la carpeta de registro, introduzca la carpeta para almacenar los registros de la interfaz de usuario.
3. En Ruta de la carpeta de registro de la consola, introduzca la carpeta para almacenar los registros de la AWS SCTinterfaz de la línea de comandos (CLI).
4. En Tamaño máximo del archivo de registro (MB), introduzca el tamaño, en MB, de un único archivo de registro. Cuando el archivo alcanza este límite, AWS SCT crea un archivo de registro nuevo.
5. En Número máximo de archivos de registro, introduzca el número de archivos de registro que se van a almacenar. Cuando el número de archivos de registro de la carpeta alcanza este límite, AWS SCT elimina el archivo de registro más antiguo.
6. En Ruta de descarga de registros de extractores, introduzca la carpeta en la que se almacenarán los registros de los agentes de extracción de datos.
7. En Ruta del registro de extractores de Cassandra, introduzca la carpeta en la que se almacenarán los registros de los agentes de extracción de datos de Apache Cassandra.
8. Seleccione Solicitar una ruta antes de cargar para asegurarse de que AWS SCT pregunta dónde almacenar los registros cada vez que utilice agentes de extracción de datos.
9. En Modo de depuración, elija True. Use esta opción para registrar información adicional cuando los registros de AWS SCT estándar no incluyan ningún problema.
10. Elija los principales módulos de aplicación para aumentar la información de registro. Puede aumentar la información de registro de los siguientes módulos de aplicación:

    - General
    - Programa de carga
    - Analizador
    - Impresora
    - Solucionador
    - Telemetría
    - Conversor
    - Asignación de tipos
    - Interfaz de usuario

- Controlador
- Comparar esquema
- Clonar centro de datos
- Analizador de aplicaciones

Para cada uno de los módulos de aplicación anteriores, elija uno de los siguientes niveles de registro:

- Seguimiento: información más detallada.
- Depuración: información detallada sobre el flujo a través del sistema.
- Información: eventos de tiempo de ejecución, como el inicio o el cierre.
- Advertencia: uso de API obsoletas, uso deficiente de la API y otras situaciones de tiempo de ejecución no deseadas o inesperadas.
- Error: errores de tiempo de ejecución o condiciones inesperadas.
- Crítico: errores que provocan el cierre de la aplicación.
- Obligatorio: el nivel de errores más alto posible.

De forma predeterminada, tras activar el Modo de depuración, AWS SCT establece el nivel de registro Información para todos los módulos de la aplicación.

Por ejemplo, para solucionar los principales problemas durante la conversión, establezca los módulos Analizador, Asignación de tipos e Interfaz de usuario en Seguimiento.

Si la información se vuelve demasiado detallada para el sistema de archivos en el que se transmiten los registros, cambie a una ubicación con espacio suficiente para capturar registros.

Para transmitir los registros a AWS Support, vaya al directorio en el que se almacenan y comprima todos los archivos en un único archivo .zip manejable. A continuación, cargue el archivo .zip con su caso de soporte. Cuando se complete el análisis inicial y se reanude el desarrollo en curso, devuelva Modo de depuración a false para eliminar el registro detallado. A continuación, aumente la velocidad de conversión.

> **Tip**
>
> Para gestionar el tamaño de los registros y agilizar la notificación de problemas, elimine los registros o muévalos a otra ubicación tras una conversión correcta. Al realizar esta tarea, se garantiza que solo se transmitan a Soporte de AWS los errores y la información pertinentes, y se evita que se llene el sistema de archivos de registro.

Solución de problemas con AWS SCT

A continuación, podrá obtener información sobre la solución de problemas con la AWS Schema Conversion Tool (AWS SCT).

No se puede cargar objetos de una base de datos de origen de Oracle

Al intentar cargar esquemas desde una base de datos de Oracle, puede que reciba uno de los siguientes errores.

```
Cannot load objects tree.
```

```
ORA-00942: table or view does not exist
```

Estos errores se producen debido a que el usuario cuyo ID se utiliza para conectarse a la base de datos Oracle no tiene permisos suficientes para leer el esquema según requiere la AWS SCT.

Puede resolver este problema si concede al usuario permiso `select_catalog_role` y permiso para cualquier diccionario en la base de datos. Estos permisos proporcionan un acceso de solo lectura a las vistas y tablas del sistema que requiere AWS SCT. En el siguiente ejemplo se crea un ID de usuario denominado `min_privs` y concede al usuario con este ID los permisos mínimos necesarios para convertir el esquema de una base de datos Oracle de origen.

```
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;
```

Mensaje de advertencia del informe de evaluación

Para evaluar la complejidad de la conversión a otro motor de base de datos, AWS SCT necesita acceder a los objetos de la base de datos de origen. Cuando AWS SCT encuentra problemas durante el análisis y no puede realizar una evaluación, se emite un mensaje de advertencia. Este mensaje indica que se ha reducido el porcentaje de conversión general. Los siguientes son los motivos por los que AWS SCT podría encontrar problemas durante el análisis:

- El usuario de la base de datos no tiene acceso a todos los objetos necesarios. Para obtener más información sobre los privilegios y permisos de seguridad requeridos por AWS SCT para su base de datos, consulte Orígenes para AWS SCT, donde encontrará la sección correspondiente a la base de datos de origen de esta guía.
- Un objeto citado en el esquema ya no existe en la base de datos. Para ayudar a resolver el problema, puede conectarse con los permisos de SYSDBA y comprobar si el objeto está presente en la base de datos.
- SCT está intentando evaluar un objeto que está cifrado.

AWS SCT Referencia CLI

En esta sección se describe cómo empezar a utilizar la interfaz de línea de AWS SCT comandos (CLI). Además, en esta sección se proporciona información sobre los comandos de teclado y los modos de uso. Para obtener una referencia completa de los comandos AWS SCT CLI, consulteMaterial de referencia.

Temas

- Requisitos previos para usar la interfaz de la línea de comandos de AWS SCT
- AWS SCT Modo interactivo CLI
- Obtener escenarios de AWS SCT CLI
- Edición de escenarios AWS SCT CLI
- AWS SCT Modo de script CLI
- AWS SCT Material de referencia CLI

Requisitos previos para usar la interfaz de la línea de comandos de AWS SCT

Descargue e instale la versión más reciente de Amazon Corretto 11. Para obtener más información, consulte Descargas para Amazon Corretto 11 en la Guía del Usuario de Amazon Corretto 11.

Descargue e instale la versión más reciente de AWS SCT. Para obtener más información, consulte ¿Instalando AWS SCT.

AWS SCT Modo interactivo CLI

Puede utilizar la interfaz de línea de AWS SCT comandos en el modo interactivo. En este modo, los comandos se introducen en la consola uno a uno. Puede usar este modo interactivo para obtener más información sobre los comandos de la CLI o descargar los escenarios de la CLI más utilizados.

Para convertir el esquema de la base de datos de origen en un esquema AWS SCT, ejecute una operación secuencial: cree un nuevo proyecto, conéctese a las bases de datos de origen y destino, cree reglas de mapeo y convierta los objetos de la base de datos. Dado que este flujo de trabajo puede ser complejo, se recomienda utilizar scripts en el modo AWS SCT CLI. Para obtener más información, consulte Modo script.

Puede ejecutar los comandos AWS SCT CLI desde la app carpeta de la ruta de AWS SCT instalación. En Windows, la ruta de instalación predeterminada es `C:\Program Files \AWS Schema Conversion Tool\`. Asegúrese de que esta carpeta incluya el archivo `AWSSchemaConversionToolBatch.jar`.

Para entrar en el modo interactivo AWS SCT CLI, utilice el siguiente comando después de completar los requisitos previos.

```
java -jar AWSSchemaConversionToolBatch.jar -type interactive
```

Ahora puede ejecutar comandos AWS SCT CLI. Asegúrese de terminar los comandos con `/` en una línea nueva. Además, debe utilizar comillas simples rectas (`'`) antes y después de los valores de los parámetros de los comandos.

> **Note**
>
> Si el comando anterior devuelve un `Unexpected error`, pruebe lo siguiente:
>
> ```
> java -Djdk.jar.maxSignatureFileSize=20000000 -jar
>  AWSSchemaConversionToolBatch.jar
> ```

Para ver la lista de comandos disponibles en el modo interactivo AWS SCT CLI, ejecute el siguiente comando.

```
help
/
```

Para ver información sobre un comando AWS SCT CLI, utilice el siguiente comando.

```
help -command: 'command_name'
/
```

En el ejemplo anterior, sustituya *`command_name`* por el nombre de un comando.

Para ver información sobre los parámetros de un comando AWS SCT CLI, utilice el siguiente comando.

```
help -command: 'command_name' -parameters: 'parameters_list'
```

```
/
```

En el ejemplo anterior, sustituya *command_name* por el nombre de un comando. A continuación, sustituya *parameters_list* por una lista de nombres de parámetros separados por una coma.

Para ejecutar un script desde un archivo en el modo interactivo AWS SCT CLI, utilice el siguiente comando.

```
ExecuteFile -file: 'file_path'
/
```

En el ejemplo anterior, sustituya *file_path* por la ruta del archivo con un script. El archivo debe tener la extensión `.scts`.

Para salir del modo interactivo AWS SCT CLI, ejecute el `quit` comando.

Ejemplos

En el siguiente ejemplo se muestra información sobre el comando `Convert`.

```
help -command: 'Convert'
/
```

En el siguiente ejemplo se muestra información sobre los parámetros del comando `Convert`.

```
help -command: 'Convert' -parameters: 'filter, treePath'
/
```

Obtener escenarios de AWS SCT CLI

Para obtener los AWS SCT escenarios más utilizados, puede usar el `GetCliScenario` comando. Puede ejecutar este comando en modo interactivo y, a continuación, editar las plantillas descargadas. Utilice los archivos editados en modo script.

El comando `GetCliScenario` guarda la plantilla seleccionada o todas las plantillas disponibles en el directorio especificado. La plantilla contiene el conjunto completo de comandos para ejecutar un script. Edite las rutas de los archivos, las credenciales de la base de datos, los nombres de los objetos y otros datos de estas plantillas. Además, asegúrese de eliminar los comandos que no utilice y de añadir nuevos comandos al script cuando sea necesario.

Para ejecutar el comando `GetCliScenario`, complete los requisitos previos y acceda al modo interactivo de la CLI de AWS SCT . Para obtener más información, consulte Modo interactivo.

A continuación, utilice la siguiente sintaxis para ejecutar el comando `GetCliScenario` y obtener los escenarios de AWS SCT .

```
GetCliScenario -type: 'template_type' -directory: 'file_path'
/
```

En el ejemplo anterior, sustituya *`template_type`* por uno de los tipos de plantilla de la tabla siguiente. A continuación, sustituya *`file_path`* por la ruta de la carpeta en la que desee descargar los scripts. Asegúrese de que AWS SCT puede acceder a esta carpeta sin solicitar derechos de administrador. Además, debe utilizar comillas simples rectas (`'`) antes y después de los valores de los parámetros de los comandos.

Para descargar todas las plantillas AWS SCT CLI, ejecute el comando anterior sin la `-type` opción.

La siguiente tabla incluye los tipos de plantillas AWS SCT CLI que puede descargar. Para cada plantilla, se incluye en la tabla el nombre del archivo y la descripción de las operaciones que se pueden ejecutar mediante el script.

Tipo de plantilla	Nombre de archivo	Descripción
BTEQ ScriptConversion	`BTEQScriptConversionTemplate.scts`	Convierte Teradata Basic Teradata Query (BTEQ) y FastLoad scripts a MultiLoad Amazon FastExport Redshift RSQL. Para obtener más información, consulte Conversión de procesos ETL.
ConversionApply	`ConversionTemplate.scts`	Convierte los esquemas de la base de datos de origen y aplica el código convertido a la base de datos de destino. Si lo desea, guarda el código convertido como un script de SQL y guarda el informe

Tipo de plantilla	Nombre de archivo	Descripción
		de evaluación. Para obtener más información, consulte Convertir esquemas de bases de datos.
GenericAppConversion	`GenericApplicationConversionTemplate.scts`	Convierte el código SQL integrado en sus aplicaciones con el conversor de aplicaciones genérico. AWS SCT Para obtener más información, consulte Conversión del código SQL de sus aplicaciones.
HadoopMigration	`HadoopMigrationTemplate.scts`	Migra sus clústeres de Hadoop en las instalaciones a Amazon EMR. Para obtener más información, consulte Uso de Apache Hadoop como origen para AWS SCT.
HadoopResumeMigration	`HadoopResumeMigrationTemplate.scts`	Reanuda una migración interrumpida del clúster de Hadoop en las instalaciones a Amazon EMR. Para obtener más información, consulte Uso de Apache Hadoop como origen para AWS SCT.

Tipo de plantilla	Nombre de archivo	Descripción
Informatica	`InformaticaConversionTemplate.scts`	Convierte el código SQL incrustado en los scripts de extracción, transformación y carga (ETL) de Informatica. Configura las conexiones a las bases de datos de origen y destino en los scripts de ETL y guarda los scripts convertidos tras la conversión. Para obtener más información, consulte Convertir los scripts de ETL de Informatica.
LanguageSpecificAppConversion	`LanguageSpecificAppConversionTemplate.scts`	Convierte el código SQL incrustado en sus aplicaciones de C#, C++, Java y Pro*C con el conversor de aplicaciones de AWS SCT . Para obtener más información, consulte Convertir el SQL de las aplicaciones.
OozieConversion	`OozieConversionTemplate.scts`	Convierte sus flujos de trabajo de Apache Oozie en AWS Step Functions. Para obtener más información, consulte Usar Apache Oozie como origen para AWS SCT.

Tipo de plantilla	Nombre de archivo	Descripción
RedshiftAgent	`DWHDataMigrationTemplate.scts`	Convierte los esquemas de almacenamiento de datos de origen y aplica el código convertido a la base de datos de Amazon Redshift de destino. A continuación, registra un agente de extracción de datos, crea e inicia una tarea de migración de datos. Para obtener más información, consulte Migración de un almacén de datos a Amazon Redshift.
ReportCreation	`ReportCreationTemplate.scts`	Crea un informe de migración de bases de datos para varios esquemas de bases de datos de origen. A continuación, guarda este informe como un archivo CSV o PDF. Para obtener más información, consulte Informes de evaluación de migración.
SQL ScriptConversion	`SQLScriptConversionTemplate.scts`	Convierte los scripts de SQL*Plus o TSQL a PL/SQL y guarda los scripts convertidos. Además, guarda un informe de evaluación.

Después de descargar la plantilla AWS SCT CLI, utilice el editor de texto para configurar el script para que se ejecute en las bases de datos de origen y destino. A continuación, utilice el modo de script AWS SCT CLI para ejecutar el script. Para obtener más información, consulte AWS SCT Modo de script CLI.

Ejemplos

En el siguiente ejemplo, se descargan todas las plantillas a la carpeta `C:\SCT\Templates`.

```
GetCliScenario -directory: 'C:\SCT\Templates'
/
```

En el siguiente ejemplo, se descarga la plantilla para la operación `ConversionApply` a la carpeta `C:\SCT\Templates`.

```
GetCliScenario -type: 'ConversionApply' -directory: 'C:\SCT\Templates'
/
```

Edición de escenarios AWS SCT CLI

Después de descargar las plantillas de escenarios, configúrelas para obtener scripts funcionales que puedan ejecutarse en sus bases de datos.

Para todas las plantillas, asegúrese de proporcionar la ruta a los controladores de las bases de datos de origen y destino. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios.

Debe incluir las credenciales de las bases de datos de origen y destino. Además, debe configurar reglas de asignación para describir un par origen-destino para su proyecto de conversión. Para obtener más información, consulte Creación de reglas de asignación.

A continuación, configure el alcance de las operaciones que se van a ejecutar. Puede eliminar los comandos que no utilice o añadir nuevos comandos al script.

Por ejemplo, supongamos que planea convertir todos los esquemas de la base de datos Oracle de origen a PostgreSQL. A continuación, tiene pensado guardar el informe de evaluación de la migración de la base de datos en formato PDF y aplicar el código convertido a la base de datos de destino. En este caso, puede utilizar la plantilla para la operación `ConversionApply`. Utilice el siguiente procedimiento para editar la plantilla AWS SCT CLI.

Para editar la plantilla AWS SCT CLI de la **`ConversionApply`** operación

1. Abra el `ConversionTemplate.scts` que descargó. Para obtener más información, consulte Ejemplos.

2. Elimine CreateFiltery convierta las operaciones ApplyToTarget -filter, -filter, SaveTargetSQLLbyStatement, SaveTargetSQ y SaveReportCSV del script.
3. Para oracle_driver_file en la SetGlobalSettingsoperación, introduzca la ruta del controlador de Oracle. A continuación, en postgresql_driver_file, introduzca la ruta al controlador de PostgreSQL.

   Si utiliza otros motores de bases de datos, utilice los nombres correspondientes para la configuración. Para obtener una lista completa de los ajustes globales que puede configurar en la SetGlobalSettingsoperación, consulte la matriz de ajustes globales del. Material de referencia
4. (Opcional) Para CreateProject, introduzca el nombre del proyecto y la ubicación del archivo de proyecto local. Si decide continuar con los valores predeterminados, asegúrese de que AWS SCT pueda crear archivos en la carpeta `C:\temp` sin solicitar derechos de administrador.
5. Para AddSource, introduzca la dirección IP del servidor de la base de datos de origen. Introduzca también el nombre de usuario, la contraseña y el puerto para conectarse al servidor de base de datos de origen.
6. Para AddTarget, introduzca la dirección IP del servidor de base de datos de destino. Introduzca también el nombre de usuario, la contraseña y el puerto para conectarse al servidor de base de datos de destino.
7. (Opcional) Para AddServerMapping, introduzca los objetos de base de datos de origen y destino que desee añadir a una regla de mapeo. Puede utilizar los parámetros `sourceTreePath` y `targetTreePath` para especificar la ruta a los objetos de la base de datos. Si lo desea, puede utilizar `sourceNamePath` y `targetNamePath` para especificar los nombres de los objetos de la base de datos. Para más información, consulte Comandos de asignación de servidor en la Material de referencia.

   Los valores predeterminados de la AddServerMappingoperación asignan todos los esquemas de origen a la base de datos de destino.
8. Guarde el archivo y, a continuación, utilice el modo script para ejecutarlo. Para obtener más información, consulte Modo script.

AWS SCT Modo de script CLI

Tras crear un script AWS SCT CLI o editar una plantilla, puede ejecutarlo con el `RunSCTBatch` comando. Guarde el archivo con el script de CLI como extensión `.scts`.

Puede ejecutar scripts AWS SCT CLI desde la app carpeta de la ruta de AWS SCT instalación. En Windows, la ruta de instalación predeterminada es `C:\Program Files \AWS Schema Conversion Tool\`. Asegúrese de que esta carpeta incluya el archivo `RunSCTBatch.cmd` o `RunSCTBatch.sh`. Además, esta carpeta debe incluir el archivo `AWSSchemaConversionToolBatch.jar`.

Como alternativa, puede añadir la ruta al archivo `RunSCTBatch` en la variable de entorno `PATH` de su sistema operativo. Tras actualizar la variable de `PATH` entorno, puede ejecutar scripts AWS SCT CLI desde cualquier carpeta.

Para ejecutar un script AWS SCT CLI, utilice el siguiente comando en Windows.

```
RunSCTBatch.cmd --pathtoscts "file_path"
```

En el ejemplo anterior, sustituya *`file_path`* por la ruta del archivo con un script.

Para ejecutar un script AWS SCT CLI, utilice el siguiente comando en Linux.

```
RunSCTBatch.sh --pathtoscts "file_path"
```

En el ejemplo anterior, sustituya *`file_path`* por la ruta del archivo con un script.

Puede proporcionar parámetros opcionales en este comando, como las credenciales de la base de datos, el nivel de detalles de la salida de la consola y otros. Para obtener más información, descargue la referencia de la interfaz de línea de AWS SCT comandos enMaterial de referencia.

Ejemplos

En el ejemplo siguiente se ejecuta el script de `ConversionTemplate.scts` en la carpeta `C:\SCT \Templates`. Puede usar este ejemplo en Windows.

```
RunSCTBatch.cmd --pathtoscts "C:\SCT\Templates\ConversionTemplate.scts"
```

En el ejemplo siguiente se ejecuta el script de `ConversionTemplate.scts` en el directorio `/ home/user/SCT/Templates`. Puede usar este ejemplo en Linux.

```
RunSCTBatch.sh --pathtoscts "/home/user/SCT/Templates/ConversionTemplate.scts"
```

AWS SCT Material de referencia CLI

Puede encontrar material de referencia sobre la interfaz de línea de AWS Schema Conversion Tool comandos (CLI) en la siguiente guía: AWS Schema Conversion Tool CLI Reference.

Notas de publicación para AWS SCT

Esta sección contiene las notas de la versión 1.0.640 AWS SCT, a partir de la versión 1.0.640.

Notas de publicación de la AWS SCT compilación 675

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversió n de AWS DMS esquemas
Cassandra	DynamoD	Se ha corregido un error que provocaba que la instalación de Cassandra fallara en el centro de datos de destino.	No
DB2 LUW	PostgreSQ L	SQL DINÁMICO: sentencia PREPARE: resolución y conversión sin SQL dinámico.	No
DB2 LUW	PostgreSQ L	Se agregó soporte para SPECIAL REGISTER.	No
DB2 LUW	PostgreSQ L	Actualización del paquete de extensión	No
Hadoop	Amazon E	Se agregó soporte para conectarse a un clúster de Hadoop mediante el protocolo rsa-sha2.	No
Microsoft SQL Server	Amazon Redshift	Se corrigió el problema de que el controlador JDBC forzara el TLS a pesar de no estar configurado.	No
Netezza	Amazon Redshift	Se agregó soporte para la conversión de vistas materializadas.	No
Oracle	Amazon Redshift	Se agregó compatibilidad con consultas recursivas en Amazon Redshift.	Sí

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
Oracle	PostgreSQL, Aurora PostgreSQL	Se corrigió una conversión incorrecta del tipo de datos NUMBER.	Sí
Oracle	Amazon Redshift	Migración de datos. Particionamiento automático de Oracle. Se agregó el tiempo de caducidad para el valor de los fragmentos de la tabla. El tiempo de caducidad es de 72 h. Cuando caduca, los fragmentos de datos se reconstruyen al crear una tarea de migración de datos.	No
Oracle	Amazon Redshift	SCT Data Extractor: se modificó el enfoque de carga de datos a Amazon Redshift. De forma predeterminada, el extractor no crea tablas escalonadas. En su lugar, una vez que todos los archivos de datos estén en el bucket de Amazon S3, el extractor los copia en la tabla de destino mediante un único comando COPY.	No
Oracle	Amazon Redshift	Se agregó la migración del tipo de datos RAW a la columna VARBYTE.	No
Oracle	PostgreSQL, Aurora PostgreSQL	Conversión multiversión	No
Oracle	PostgreSQL	Se agregó compatibilidad con la declaración MERGE en PostgreSQL 15.x.	Sí

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
Oracle	PostgreSQL	Se agregó soporte para nuevas funciones de expresiones regulares en PostgreSQL 15.x.	Sí
Oracle	PostgreSQL, Aurora PostgreSQL	La sentencia ON CONFLICT DO UPDATE se convierte sin excluir el alias.	Sí
Teradata	Amazon Redshift	Se ha añadido soporte de conversión para las funciones LEAD/LAG.	No
Teradata	Amazon Redshift	Se ha mejorado la conversión de tipos de datos con indicación explícita del formato de los datos.	No
Teradata	Amazon Redshift	Se ha mejorado la conversión de la cláusula AT «TIME ZONE» en expresiones de hora y hora.	No
Teradata	Amazon Redshift	AI-9996 durante los procedimientos de conversión con sentencias MERGE.	No

Notas de publicación de la compilación 674 AWS SCT

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
Todos	Todos	Correcciones de varios errores y mejoras de rendimiento	Parcial (solo para los pares de origen y destino compatibles)

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
Azure SQL / Microsoft SQL Server	Amazon Redshift	Eliminación del elemento de acción 18066 “No se puede convertir el nombre del esquema”, que confundía a los usuarios durante la evaluación o conversión del esquema	No
Azure SQL / Microsoft SQL Server	Amazon RDS para MySQL / Amazon Aurora MySQL	Conversión incorrecta del procedimiento sin asignar un código de retorno	Parcial (la conversión de esquemas no admite actualmente Azure SQL como origen)
Azure SQL / Microsoft SQL Server	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Corrección del elemento de acción 9997 para algunos casos de conversión de cláusulas FOR XML PATH	Parcial (la conversión de esquemas no admite actualmente Azure SQL como origen)

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
Azure SQL / Microsoft SQL Server	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Redondeo del valor a la escala original en el cuerpo del procedimiento/función	Parcial (la conversión de esquemas no admite actualmente Azure SQL como origen)
Azure SQL / Microsoft SQL Server	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Varias mejoras en la conversión de instrucciones EXECUTE	Parcial (la conversión de esquemas no admite actualmente Azure SQL como origen)
Azure SQL/ Microsoft SQL Server/ Azure Synapse	Amazon Redshift	Mejora de la conversión de las siguientes instrucciones y modos: • EXCEPTION BLOCK • AUTOCOMMIT • NONATOMIC • GROUPING SET • CUBE • ROLLUP	No

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
DB2 LUW	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Diversas correcciones en sql-queries de carga de metadatos	No
DB2 LUW	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	No se espera el elemento de acción 9996 en los activadores	No
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Función analítica ROWNUMBER	No

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Compatibilidad constante con cadenas hexadecimales	No
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Diversas correcciones en sql-queries de carga de metadatos.	No
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Compatibilidad con la referencia de secuencia NEXT VALUE FOR	No

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Compatibilidad con la opción DB2_NUMBER_ROWS de la instrucción GET DIAGNOSTICS	No
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Varias instrucciones GET DIAGNOSTICS	No
DB2 z/OS	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Se corrigieron errores en la conversión de la instrucción FOR.	No

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
Oracle	Amazon RDS para MySQL / Amazon Aurora MySQL	Se corrigió un error que provocaba que el nodo de parámetros de la función del paquete no estuviera definido.	Sí
Oracle	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Se corrigieron errores en la función AWS_ORACLE_EXT.NEXT_DAY del paquete de extensión	Sí
Oracle	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Se corrigieron varios errores relacionados con la conversión «(+)» en las uniones exteriores de Oracle	Sí
Oracle		Compatibilidad con la autenticación Kerberos	No

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
SAP ASE	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Se corrigió un error al convertir más de un identificador en la cláusula FROM de la instrucción UPDATE	No
SAP ASE	Amazon RDS para PostgreSQL / Amazon Aurora PostgreSQL	Se corrigió un error relacionado con la conversión de instrucciones y comentarios de varias líneas	No
SAP ASE		Se agregó compatibilidad para el parámetro ENCRYPT_PASSWORD al conectarse	No
Teradata	Amazon Redshift	Se mejoró la conversión de la tabla VOLATILE con un nombre de esquema especificado	No
Teradata	Amazon Redshift	Conversión incorrecta de WHERE CLAUSE en un CTE complejo	No
Teradata	Amazon Redshift	Se agregó compatibilidad con el tipo de datos INTERVAL al migrar datos mediante agentes de extracción de datos de SCT.	No

Origen	Destino	Novedades, mejoras o correcciones	Disponibilidad en la conversión de AWS DMS esquemas
Scripts de BTEQ de Teradata	Scripts de RSQL de Amazon Redshift	Parámetros de conversión incorrectos en el procedimiento ejecutado por BTEQ	No

Notas de publicación de la AWS SCT compilación 673

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Correcciones de errores generales y mejoras de rendimiento
Azure SQL / Microsoft SQL Server	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se corrigió la conversión incorrecta de llamadas a funciones
Azure SQL / Microsoft SQL Server	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se implementó la conversión de la cláusula XML de `FOR`
Azure SQL / Microsoft	Aurora PostgreSQL / Amazon RDS	Conversión de una cláusula XML con un alias incorrecto de `FOR`.

Origen	Destino	Novedades, mejoras o correcciones
SQL Server	PostgreSQL	
Azure SQL / Microsoft SQL Server	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se ha corregido un error que provocaba que AWS SCT no se `EXECUTE` convirtieran sentencias que ejecutaban una cadena de caracteres con parámetros de procedimiento.
Azure SQL / Microsoft SQL Server	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se mejoró la conversión de instrucciones UPDATE con combinaciones interiores.
Azure Synapse	Amazon Redshift	Se corrigió la conversión incorrecta de la función `OBJECT_ID` integrada.
IBM DB2 para z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se implementó la conversión de las siguientes instrucciones y objetos: • `DECLARE TEMPORARY TABLE statement` • `DROP TABLE statement` • Restricciones de `PK` y `UNIQUE` en las tablas particionadas • Función `TIMESTAMPDIFF` • Función `TO_DATE` • Función `EBCDIC_STR` • Función `VARCHAR_FORMAT`

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se corrigió un error que se producía cuando el índice basado en funciones omitía las funciones después de la conversión.
IBM DB2 para z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se corrigió un error que provocaba que la instrucción `REPEAT` se cerrara con el elemento de acción 9996 después de la conversión
IBM DB2 para z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se corrigió un error que provocaba que la instrucción `FINAL TABLE` se cerrara con el mensaje 9996.
IBM DB2 para z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	`LOADER` | Clave de particionamiento en la restricción de referencias. AWS SCT ahora puede convertir las claves principales y las restricciones únicas de las tablas particionadas como índices secundarios.
IBM DB2 para z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se añadió compatibilidad con la función PostgreSQL.VARCHAR_FORMAT

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se implementó el cambio de intercalación en los comandos `CreateTransformationRule` y `ModifyTransformationRule` de la CLI de SCT.
Greenplum	Amazon Redshift	Se corrigió un error que provocaba una llamada incorrecta al procedimiento almacenado después de la conversión
Hadoop	Amazon EMF	Se agregó compatibilidad para conectarse a un clúster de Hadoop mediante el protocolo rsa-sha2.
Hadoop	Amazon EMF	Se agregó compatibilidad con Amazon EMR con un almacén de metadatos de Hive que no es de Glue,
Oracle	Amazon Redshift	Se corrigió un error que provocaba una conversión incorrecta de una consulta recursiva en la que la columna PRIOR no estaba en la lista SELECT.
Oracle	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se implementó la devolución de un elemento de una matriz asociativa
Oracle	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se corrigió el elemento de acción 9996 inesperado en `UNPIVOT` entre corchetes

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se corrigió el elemento de acción 9996 inesperado en `UNPIVOT` con `UNION ALL`
Oracle	Aurora PostgreSQL / Amazon RDS PostgreSQL	Se mejoraron las conversiones de los tipos de datos de `Number`
Oracle	Extractor de datos de Amazon Redshift	Se añadió compatibilidad para el particionamiento automático de tablas de Oracle. Optimización para crear tareas de migración.
Teradata	Amazon Redshift	Se implementó la conversión de la instrucción `EXCEPTION BLOCK`
Teradata	Amazon Redshift	Compatibilidad para la conversión de predicados `ALL`, `ANY` y `SOME` a Amazon Redshift.
Teradata	Amazon Redshift	Se agregó compatibilidad nativa para predicados `QUALIFY`.
Teradata	Amazon Redshift	Se mejoró la conversión de lo siguiente: • Consultas recursivas • `GROUPING SET` • `CUBE` • `ROLLUP` • Instrucción `UPDATE` con unión implícita

Origen	Destino	Novedades, mejoras o correcciones
Fuentes de OLAP	Extractor de datos de Amazon Redshift	Se implementaron comandos de CLI para las tareas Stop/Resume del Extractor de datos de Amazon Redshift.
Fuentes de OLAP	Extractor de datos de Amazon Redshift	Se agregó la posibilidad de seleccionar las columnas de la tabla que deben migrarse durante la configuración de la tarea de migración.

Notas de publicación de la AWS SCT compilación 672

Origen	Destino	Novedades, mejoras o correcciones
Todos	Amazon RDS para PostgreSQL	Se implementó la compatibilidad con la versión principal 15 de PostgreSQL como destino de migración.
Todos	Amazon Redshift	Se agregó un nuevo `PrintTaskStatus` comando en la interfaz de línea de AWS SCT comandos (CLI) para mostrar el estado de la tarea de migración de datos.
Todos	Amazon Redshift	Se mejoró el flujo de configuración de los agentes de extracción de datos.
Todos	Amazon Redshift	Se solucionó un error que provocaba que los agentes de extracción de datos no mostraran la información sobre las tareas secundarias.
Apache Oozie	AWS Step Functions	Se agregó una opción para guardar las definiciones de las máquinas de estado como un script en el código convertido.
Base de datos SQL Azure	Aurora PostgreSQL	Se implementó la conversión de las funciones `COALESCE`, `DATEADD`, `GETDATE` y `SUM`.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	PostgreSQL	
Base de datos SQL Azure	Aurora PostgreSQL	Se mejoró la conversión de instrucciones `UPDATE` con cláusulas `JOIN` y `OUTPUT`.
Microsoft SQL Server	PostgreSQL	
Base de datos SQL Azure	Aurora PostgreSQL	Se resolvió un error que se producía durante la conversión de la instrucción `SELECT TOP 1 WITH TIES`.
Microsoft SQL Server	PostgreSQL	
Base de datos SQL Azure	Aurora PostgreSQL	Se resolvieron varios problemas que se producían durante la conversión de las `FOR XML` cláusulas en las funciones integradas.
Microsoft SQL Server	PostgreSQL	
Greenplum	Amazon Redshift	Se implementó la conversión de instrucciones `GET DIAGNOSTICS` y `RAISE` mediante un bloque nativo `EXCEPTION` de Amazon Redshift.
Greenplum	Amazon Redshift	Se mejoró la conversión de los procedimientos almacenados al agregar compatibilidad con un bloque `EXCEPTION` en el código convertido.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se corrigió un error que provocaba la corrección incorrecta de la función `TO_CHAR` con plantillas de formato de hora.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de expresiones de tablas anidadas.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de las instrucciones `GOTO`, `MERGE`, `REPEAT` y `SIGNAL`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de instrucciones `FETCH` con palabras clave de orientación `BEFORE` y `AFTER`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de las referencias a las tablas `FINAL TABLE` y `OLD TABLE`.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de las siguientes funciones. • ADD_MONTHS • DAY con parámetros del tipo de datos de caracteres • DAYOFWEEK • DAYS • DECODE • HOUR • LAST_DAY • LOCATE_IN_STRING • MICROSECOND • MINUTE • MONTH • ROUND • TIME • TIMESTAMP • TIMESTAMP_FORMAT • TRANSLATE • UNICODE_STR • XMLCAST • XMLELEMENT • XMLQUERY • XMLSERIALIZE • YEAR
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de un alias de una subconsulta en cláusulas JOIN.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de funciones `COALESCE`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de índices `EXPLICIT`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de los nombres de las columnas en expresiones compuestas para resolver un problema que provocaba la aparición inesperada del elemento de acción 9997 durante la conversión.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de las claves principales y las restricciones únicas.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de instrucciones `XMLTABLE` en instrucciones `INSERT` para resolver un problema que provocaba la aparición inesperada del elemento de acción 9996 durante la conversión.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9996 durante la conversión de funciones con el argumento `SUBSTR`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9996 durante la conversión del registro especial `CURRENT TIMESTAMP` .
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9996 durante la conversión de instrucciones `MERGE`, instrucciones no compatibles y funciones integradas no compatibles.
Microsoft SQL Server	Todos	Se añadió compatibilidad con la versión 2022 de Microsoft SQL Server como origen.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de `SELECT` sentencias que utilizan operadores de concatenación de cadenas. AWS SCT usa la `STRING_AGG` función en el código convertido.
Microsoft SQL Server	Babelfish para Aurora PostgreSQL	Se implementó compatibilidad con la nueva versión 3.1.0 del archivo de configuración de características de Babelfish. En este archivo se definen las características de SQL que son compatibles y no compatibles con versiones específicas de Babelfish.

Origen	Destino	Novedades, mejoras o correcciones
Netezza	Amazon Redshift	Se resolvió un problema que impedía que los agentes de extracción de datos iniciaran la migración de datos desde el punto CDC especificado.
Oracle	Todos	Se actualizó el informe de evaluación de la versión 19 de las bases de datos de Oracle como origen.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la conversión `DBMS_OUTPUT` del paquete añadiendo nuevas funciones al paquete AWS SCT de extensión.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de funciones y procedimientos que utilizan matrices asociativas como argumentos o parámetros.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de cláusulas `DISTINCT` en instrucciones `SELECT`.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de tablas en las que la restricción de clave principal tiene el mismo nombre que la tabla.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión del procedimiento `RAISE_APPLICATION_ERROR` con el tercer parámetro.
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que impedía que la regla de migración cambiara automáticamente el tipo de datos `NUMERIC` a `INTEGER` cuando correspondía.
Oracle DW	Amazon Redshift	Se implementó la compatibilidad con cláusulas `CONNECT BY` nativas de Amazon Redshift en el código convertido.
Oracle DW	Amazon Redshift	Se mejoró la migración de datos al agregar automáticamente una tarea secundaria para cada tabla o partición del ámbito de la migración. Este enfoque evita la pérdida de datos en el caso de los datos insertados después del particionamiento.
Teradata	Amazon Redshift	Se implementó la conversión de vistas recursivas.
Teradata	Amazon Redshift	Se mejoró la conversión de los procedimientos almacenados que utilizan los modos de transacción `BTET` y `ANSI` al agregar compatibilidad con el modo de transacción `AUTOCOMMIT` nativo de Amazon Redshift.
Teradata	Amazon Redshift	Se mejoró la conversión de los procedimientos almacenados que utilizan la semántica de transacción `TERADATA` al agregar la palabra clave `NONATOMIC` en el código convertido.
Teradata	RSQL de Amazon Redshift	Se resolvió un problema por el que el código convertido incluía el ID de la clave de AWS acceso y la clave de acceso secreta.

Notas de la versión de la AWS SCT compilación 671

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se ha corregido un error por el que AWS SCT no tenía permisos para guardar un archivo de proyecto en Windows.
Todos	Todos	Se actualizaron las siguientes AWS SCT plantillas de interfaz de línea de comandos (CLI). • BTEQ ScriptConversion • ConversionApply • HadoopMigration • HadoopResumeMigration • Informatica Para obtener más información sobre las plantillas AWS SCT CLI, consulteObtención de escenarios de la CLI.
Todos	Amazon Redshift	Se corrigió un error por el que AWS SCT no se creaba un paquete de extensión en la interfaz de línea de comandos (CLI).
Todos	Amazon Redshift	Se resolvió un problema por el que los agentes de extracción de AWS SCT datos no utilizaban la AWS Snowball configuración de la interfaz de línea de comandos (CLI).
Apache Oozie	AWS Step Functions	Se implementó el soporte para la migración de Apache Oozie al AWS Step Functions modo de interfaz de línea de comandos (CLI). Tras migrar sus cargas de trabajo de Hadoop a Amazon EMR, ahora puede migrar el sistema de programación del flujo de trabajo a Nube de AWS. Para obtener más información, consulte Conversión de Apache Oozie a AWS Step Functions .
Base de datos SQL Azure	Aurora PostgreSQL	Se corrigió un error del solucionador que se producía en las tablas y los alias.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	PostgreSQL	
Base de datos SQL Azure	Aurora PostgreSQL	Se implementó la conversión de cláusulas `INDEX ON`.
Microsoft SQL Server	PostgreSQL	
Base de datos SQL Azure	Aurora PostgreSQL	Se mejoró la conversión de los siguientes objetos para evitar elementos de acción inesperados. • Instrucciones por lotes • Listas de expresiones • Alias de tabla • Tablas temporales • Desencadenadores • Variables de usuario
Microsoft SQL Server	PostgreSQL	
Base de datos SQL Azure	Aurora PostgreSQL	Se resolvió un error de análisis que se producía en los procedimientos.
Microsoft SQL Server	PostgreSQL	

Origen	Destino	Novedades, mejoras o correcciones
Base de datos SQL Azure Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se ha corregido un error que provocaba que AWS SCT se utilizaran nombres incorrectos de las tablas temporales en el código convertido para `OBJECT_ID` las funciones.
Base de datos SQL Azure Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvieron problemas que provocaban la aparición inesperada del elemento de acción 9996 durante la conversión de los siguientes elementos de código. • Funciones `CONVERT` • Funciones `DATEADD` • Instrucciones `DELETE` dentro de funciones en línea • Instrucciones `IF` • Acciones `INSERT` o `UPDATE` en una columna • Instrucciones `RETURN` • Instrucciones `UPDATE` con consultas o funciones complejas
BigQuery	Amazon Redshift	Se agregó soporte BigQuery como fuente para el proceso de evaluación multiservidor. Para obtener más información, consulte Crear un informe de evaluación multiservidor.
Hadoop	Amazon EMF	Se actualizó la versión del controlador JDBC Apache Hive compatible que utiliza para conectarse a sus bases de datos de origen. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se ha mejorado el cargador de metadatos de origen para garantizar que AWS SCT carga los objetos de la base de datos de origen, como las claves principales, los índices implícitos, etc.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se corrigió un error del solucionador que se producía en las columnas de los cursores implícitos.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la habilidad de conservar el formato de los nombres de las columnas, las expresiones y las cláusulas de las instrucciones DML en el código convertido.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de claves externas de esquemas cruzados.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de las funciones `LENGTH` y `VARCHAR`.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de instrucciones `LABEL ON` y `DECLARE CONDITION` .
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de instrucciones `SELECT` con cláusulas `OPTIMIZE FOR`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de instrucciones `CREATE TABLE` al agregar valores predeterminados para todos los tipos de datos compatibles.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de atributos `INCREMENT BY`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de tablas particionadas al agregar la capacidad de excluir las particiones de las tablas del ámbito de la conversión.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de las definiciones de clave principal con columnas `INCLUDE`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de la función `SUBSTRING` .
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de instrucciones `SET` y `DECLARE HANDLER FOR`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de tipos de datos variables.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de funciones `XMLTABLE`.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró el flujo de migración al implementar el siguiente orden de aplicación de los objetos convertidos a la base de datos de destino: tablas, particiones, índices, restricciones, claves externas y desencadenadores.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9996 durante la conversión de comentarios en el código fuente.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9997 durante la conversión del alias en cláusulas `FROM`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9997 durante la conversión del alias del cursor.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se corrigió un error que provocaba que el código convertido devolvier a resultados diferentes para instrucciones `SELECT` con cláusulas `ORDER BY`. Como SQL Server y PostgreSQL tratan los valores `NULL` de forma diferente, el código convertido ahora incluye cláusulas `NULLS FIRST` o `NULLS LAST` que garantizan que el código convertido devuelva los resultados en el mismo orden que el código fuente.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de tipos de datos en funciones de tabla.
MySQL	Amazon RDS para MySQL	Se resolvió un problema que provocaba la aparición inesperada de comillas simples (`' '`) alrededor de los nombres de objeto de la base de datos en el código convertido.
Oracle	Aurora PostgreSQL PostgreSQL	Se agregaron nuevas vistas al paquete de extensión para simular las vistas del sistema Oracle que se utilizan para mostrar información sobre particiones y subparticiones.
Oracle	Aurora PostgreSQL PostgreSQL	Se actualizaron dos funciones del paquete de extensión para agregar nombres de esquema como argumentos en el código convertido.
Oracle	Aurora PostgreSQL PostgreSQL	Se ha corregido un error que AWS SCT provocaba que no se utilizaran los parámetros correctos para la conversión de aplicaciones de C++ tras actualizar el código de la aplicación en la interfaz de usuario.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de instrucciones `CREATE TYPE` para evitar excepciones inesperadas.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de tablas anidadas.
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un error de análisis que se producía en los objetos del paquete.
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un problema por el que se recortaban AWS SCT inesperadamente los nombres de los objetos en el código convertido cuando la longitud del nombre superaba los 60 caracteres.
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de los desencadenadores de nivel de fila de las tablas particionadas.

Origen	Destino	Novedades, mejoras o correcciones
Oracle DW	Amazon Redshift	Se implementó compatibilidad de particionamiento automático de tablas para la migración de datos. Para acelerar la migración de datos, AWS SCT puede particionar automáticamente tablas o particiones grandes en función de los valores de la `ROWID` pseudocol umna. Para obtener más información, consulte Uso de particiones nativas.
Teradata	Amazon Redshift	Se implementó la compatibilidad con cláusulas `MERGE` nativas de Amazon Redshift en el código convertido. Para obtener más información sobre el comando `MERGE` en Amazon Redshift, consulte MERGE en la Guía para desarrolladores de bases de datos de Amazon Redshift.
Teradata	Amazon Redshift	Se mejoró la conversión de instrucciones `DELETE` y `UPDATE` que no utilizan nombres de tablas explícitos.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de instrucciones `IN` y `NOT IN`.

Notas de publicación de la AWS SCT compilación 670

Origen	Destino	Novedades, mejoras o correcciones
Base de datos SQL Azure	Aurora PostgreSQ L	Se resolvieron problemas que provocaban la aparición inesperada del elemento de acción 9996 durante la conversión de los siguientes elementos de código.
Microsoft SQL Server	PostgreSQ L	• Instrucciones `CREATE INDEX` dentro de instrucciones `INCLUDE` • Instrucciones `DECLARE` • Instrucciones `DECLARE ... TABLE` • `DECLARE` con valores predeterminados dentro de instrucciones `LOOP` • Instrucciones `DELETE`

Origen	Destino	Novedades, mejoras o correcciones
		• Instrucciones `DROP CONSTRAINT` dentro de instrucciones `ALTER TABLE` • `EXECUTE AS CALLER` y `REVERT` • Instrucciones `IIF` • Listas de expresiones • Funciones `MONTH()` • Instrucciones `UPDATE` • Funciones `YEAR()`
Azure Synapse Analytics	Amazon Redshift	Se añadió compatibilidad con Azure Synapse Analytics como origen para el proceso de evaluación multiservidor. Para obtener más información, consulte Crear un informe de evaluación multiservidor.
Hadoop	Amazon EMF	Se implementó compatibilidad para la migración de clústeres de Hadoop a Amazon EMR en el modo de interfaz de la línea de comandos (CLI). Para obtener más información, consulte Migración de marcos de macrodatos.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se corrigió un error del solucionador que se producía en las tablas y los alias.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de expresiones `CASE`.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de referencias `CURRENT_DATE` a registros especiales. Una referencia a un registro especial en Db2 para z/OS es una referencia a un valor proporcionado por el servidor actual.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de las funciones `DATE` y `POSSTR`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de las constantes de fecha y hora.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de los valores predeterminados de las columnas de los siguientes tipos de datos: `DATE`, `TIME`, `TIMESTAMP` y `TIMESTAMP WITH TIME ZONE`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9996 durante la conversión de instrucciones `SELECT INTO`.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de funciones `DATEDIFF`.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se corrigió un error que provocaba la conversión de funciones `ISNULL` a `NULLIF`. Como resultado, el código convertido producía resultados diferentes en comparación con el código fuente. Ahora, AWS SCT convierte `ISNULL` las funciones en`COALESCE`.
Netezza	Amazon Redshift	Se mejoraron los agentes de extracción de datos para resolver un problema en el que se establecía el estado de error para las tareas que se habían completado correctamente.
Netezza	Amazon Redshift	Se añadió la posibilidad de cambiar los puntos de conexión de las tareas secundarias tras iniciar una migración de datos con agentes de extracción de datos.
Microsoft SQL Server MySQL Oracle PostgreSQL	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Se agregó la capacidad de conectarse a bases de datos mediante un protocolo de direcciones IPv6.
Oracle	Amazon RDS para Oracle	Se implementó la conversión del paquete `DBMS_JOB` que programa y administra los trabajos en la cola de trabajos.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se agregaron nuevas funciones al paquete de extensión para mejorar la conversión de las tablas anidadas globales. Estas nuevas funciones simulan las funciones `DELETE`, `EXTEND` y `TRIM` en su código fuente de Oracle.
Oracle	Aurora PostgreSQL PostgreSQL	Se añadió la posibilidad de especificar el ámbito de la conversión del código SQL incrustado en las aplicaciones de Java. Ahora puede excluir del ámbito de la conversión los subconjuntos del proyecto de la aplicación de origen. Para obtener más información, consulte Conversión del código SQL de la aplicación de Java en AWS SCT.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de los operadores de concatenación (`||`) dentro de los índices funcionales.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de condiciones `IN` en las que el código fuente no incluye paréntesis para una sola expresión.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de instrucciones `MERGE` a `INSERT ON CONFLICT` en PostgreSQL.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un error de análisis que se producía en los paquetes de procedimientos.
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 5072 durante la conversión de paquetes.
Oracle DW	Amazon Redshift	Se ha corregido un error por el que AWS SCT no se aplicaba el paquete de extensiones al aplicar el código convertido a la base de datos de destino.
Oracle DW	Amazon Redshift	Se ha corregido un error AWS SCT que impedía aplicar algunos de los archivos del paquete de extensiones al utilizar el asistente del paquete de extensiones.
Oracle DW	Amazon Redshift	Se resolvió un problema por el que no se AWS SCT podía procesar la migración de datos AWS Snowball con más de 500 tareas ejecutándose en paralelo.
Oracle DW	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de las funciones definidas por el usuario con tipos definidos por el usuario.

Notas de la versión de la AWS SCT compilación 669

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se mejoró el proceso de evaluación multiservidor, lo que ayuda a determinar la plataforma de base de datos de destino óptima para

Origen	Destino	Novedades, mejoras o correcciones
		las bases de datos de origen. Ahora, AWS SCT ignora la AWS Secrets Manager clave si proporciona las credenciales de la base de datos en el archivo de valores separados por comas (CSV) de entrada. Para obtener más información, consulte Crear un informe de evaluación multiservidor.
Todos	Todos	Se ha resuelto un problema por el que el informe de evaluación multiservidor incluía la dirección IP de la base de datos de origen cuando se utilizaba un identificador secreto AWS Secrets Manager para conectarse a la base de datos.
Todos	Amazon Redshift	Se implementó la configuración automática de los ajustes de la máquina virtual Java (JVM) en función del sistema operativo y de la RAM disponible. AWS SCT utiliza esta JVM para ejecutar el trabajo de los agentes de extracción de datos.
Todos	Amazon Redshift	Se resolvió un problema que impedía el inicio de los agentes de extracción de datos en Ubuntu.
Todos	Amazon Redshift	Se resolvió un problema que impedía el inicio de las tareas de extracción de datos después de ejecutar el archivo `StartAgent.bat` en Windows.
Base de datos SQL Azure Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de los nombres de las columnas con la opción Generar nombres únicos para los índices activada.
Greenplum	Amazon Redshift	Se implementó la conversión de las funciones que devuelven `VOID` a los procedimientos.

Origen	Destino	Novedades, mejoras o correcciones
Greenplum	Amazon Redshift	Se resolvió un problema por el que la migración de datos fallaba cuando la base de datos de origen incluía valores no numéricos (NaN) en las columnas numéricas. AWS SCT los agentes de extracción de datos ahora sustituyen los valores de NaN por NULL.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se agregó una nueva configuración de conversión para especificar las opciones `DATE FORMAT` y `TIME FORMAT` durante la conversión de funciones `CHAR` integradas.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se agregó un elemento de acción 8534 para la conversión de cursores predefinidos que se declaraban con la cláusula. `WITHOUT RETURN`. Si el cursor no devuelve conjuntos de resultados, AWS SCT asigna un `NULL` valor al nombre del cursor en el código convertido y genera un elemento de acción.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se editó la propiedad `CURRENT CLIENT_APPLNAME` que identifica AWS SCT durante la conexión a la base de datos de origen.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó una nueva configuración de conversión para especific ar las opciones `DATE FORMAT` y `TIME FORMAT` durante la conversió n de funciones `CHAR` integradas.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de instrucciones `LEAVE` en instrucciones `BEGIN...END` en bloque.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de las funciones `XMLPARSE`, `XMLTABLE` y `XMLNAMESPACES` .
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de las funciones `CHAR` integradas.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de cursores.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9996 durante la conversión de instrucciones `FOR` en bucle.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión del uso de los tipos de tablas en instrucciones `SELECT`.
Microsoft SQL Server	Babelfish para Aurora PostgreSQL	Se implementó la compatibilidad para la nueva versión 2.2.0 del archivo de configuración de características de Babelfish. En este archivo se definen las características de SQL que son compatibles y no compatibles con versiones específicas de Babelfish.
Netezza	Amazon Redshift	Se mejoraron los agentes de extracción de datos para resolver un problema que impedía la eliminación de una fila de la tabla de destino durante la replicación de datos en curso.
Oracle	Amazon RDS para Oracle	Se mejoró la conversión de las funciones de Oracle Database Enterprise Edition.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de funciones `GROUPING_ID` .
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de código SQL en aplicaciones de C# al agregar compatibilidad con la asignación personalizada de tipos de datos en el modo de interfaz de la línea de comandos (CLI).

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de tablas anidadas para evitar un elemento de acción 9996 inesperado.
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de la llamada de un constructor de objetos.
Oracle DW	Amazon Redshift	Se implementó compatibilidad de particionamiento de tablas existentes para la migración de datos. Para acelerar la migración de datos, AWS SCT crea subtareas para cada partición de la tabla de origen que no esté vacía. Para obtener más información, consulte Uso de particiones nativas.
Teradata	Amazon Redshift	Se mejoró la conversión de funciones `CAST` con argumentos `TIME WITH TIME ZONE AS TIMESTAMP` , `TIME WITH TIME ZONE AS CHAR` y `TIMESTAMP AS TIME WITH TIME ZONE`.
Teradata	Amazon Redshift	Se mejoró la conversión de funciones `CAST` con la opción `FORMAT`.
Teradata	Amazon Redshift	Se resolvió un problema que impedía la conversión de funciones `CEIL`.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de instrucciones `MERGE` con cláusulas `DELETE`.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de funciones `TO_CHAR` con argumentos de fecha y formato.

Notas de publicación de la AWS SCT compilación 668

Origen	Destino	Novedades, mejoras o correcciones
Todos	Amazon Redshift	Se resolvió un problema que provocaba el funcionamiento incorrect o de operadores de multiplicación en las reglas de migración. Estos operadores permiten cambiar la longitud de tipos de datos `char`, `varchar`, `nvarchar` y `string`. Para obtener más información, consulte Crear reglas de migración.
Azure Synapse Analytics	Amazon Redshift	Se implementó la compatibilidad de funciones `CONVERT` con argumentos `VARCHAR`.
Azure Synapse Analytics	Amazon Redshift	Se mejoró la conversión de instrucciones `SELECT` con cláusulas `NOLOCK` .
Azure Synapse Analytics	Amazon Redshift	Se mejoró la conversión de instrucciones `UPDATE` con alias o con cláusulas `SET` y `FROM`.
Greenplum	Amazon Redshift	Se implementó el particionamiento virtual automático para la migración de datos. AWS SCT utiliza la columna del sistema `GP_SEGMENT_ID` para crear particiones.
Greenplum	Amazon Redshift	Se implementó la compatibilidad de cláusulas `RETURN QUERY` y `RETURN SETOF`.
Greenplum	Amazon Redshift	Se implementó la compatibilidad de funciones `SUBSTRING` con tres parámetros.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de funciones `SUBSTR` con parámetros `LOCATE`.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se agregó una opción para usar una matriz de variables `REFCURSOR` para devolver conjuntos de resultados dinámicos. Al seleccionar esta opción en la configuración de conversión, AWS SCT agrega un parámetro `OUT` adicional en el código convertido.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la compatibilidad de sentencias `FOR` en bucle.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se implementó la compatibilidad de funciones `XMLPARSE`. Se agregó un elemento de acción 8541 para la división de espacios en blanco en funciones `XMLPARSE`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de varios controladores de excepciones en un único bloque `BEGIN ... END`.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de los desencadenadores `INSERT` y `DELETE`.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de llamadas a procedimientos anidados.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de tipos de tablas.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de operaciones `NOT` lógicas bit a bit en valores enteros.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que impedía inicializar las matrices locales en PostgreSQL 8.0.2 y versiones anteriores.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de instrucciones `MERGE` con cláusulas `WHEN NOT MATCHED BY SOURCE`.

Origen	Destino	Novedades, mejoras o correcciones
MySQL	Aurora MySQL	Se resolvió un problema por el que se determinaban AWS SCT incorrectamente los permisos de usuario que concedía el `rds_superuser_role` rol.
Netezza	Amazon Redshift	Se ha mejorado el cargador de metadatos de origen para garantizar que se carguen AWS SCT correctamente los objetos de la base de datos con los nombres en minúsculas.
Oracle	Aurora PostgreSQL PostgreSQL	Se agregaron nuevas funciones al paquete de extensión para mejorar la conversión de las tablas anidadas globales. Estas nuevas funciones simulan las funciones `PRIOR`, `NEXT`, `LIMIT`, `FIRST`, `LAST`, `EXISTS`, `EXTEND`, `TRIM`, `DELETE` y `SET` en su código fuente de Oracle. Para obtener más información, consulte Uso de paquetes de extensión.
Oracle	Aurora PostgreSQL PostgreSQL	Se añadió la posibilidad de especificar el ámbito de la conversión para aplicaciones de C#. Los usuarios ahora puede excluir del ámbito de la conversión los subconjuntos del proyecto de la aplicación de origen.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la compatibilidad de métodos `COUNT` en las colecciones.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la compatibilidad de variables y constructores en tablas anidadas.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la compatibilidad de funciones `RATIO_TO_REPORT` y `STANDARD_HASH` .
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de objetos grandes (LOB) como parte del paquete de extensión de AWS SCT .
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de colecciones locales.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de instrucciones `JOIN` con cláusulas `USING`en las que los nombres de las columnas no incluyen el nombre de la tabla.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de las funciones `EMPTY_BLOB` y `EMPTY_CLOB` .

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de variables de enlace posicional en aplicaciones de C#.
SAP ASE	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de desencadenadores de eventos múltiples.
SAP ASE	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de desencadenadores recursivos.
SAP ASE	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de desencadenadores con la variable global `@@rowcount` .
SAP ASE	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de las funciones de agregación de la cláusula `SET` de instrucciones `UPDATE`.

Origen	Destino	Novedades, mejoras o correcciones
SAP ASE	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 42702 durante la conversión de instrucciones `UPDATE`.
SAP ASE	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de funciones `CONVERT` con argumentos `CHAR`.
Snowflake	Amazon Redshift	Se ha añadido la compatibilidad con Snowflake como fuente para la migración de datos con AWS SCT agentes de extracción de datos. Para obtener más información, consulte Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift.
Teradata	Amazon Redshift	Se mejoró la conversión de funciones `CAST` con argumentos `TIMESTAMP AS TIME WITH TIMEZONE`.

Notas de la versión de la compilación AWS SCT 667

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se implementó la compatibilidad con los scripts de extracción, transformación y carga (ETL) de Informatica en el modo de interfaz de línea de comandos (CLI). AWS SCT redirige automáticamente los scripts ETL de Informatica a la nueva base de datos de destino. Además, AWS SCT convierte los nombres de los objetos y el código SQL incrustado en los objetos de Informatica. Para obtener más información, consulte Convertir los scripts de ETL de Informatica.

Origen	Destino	Novedades, mejoras o correcciones
Todos	Amazon Redshift	Se aumentó la versión mínima del controlador admitida para Amazon Redshift a 2.1.0.9. Para obtener más información, consulte Descarga de los controladores de base de datos necesarios.
Azure Synapse Analytics	Amazon Redshift	Se agregó una nueva función al paquete de extensión para mejorar la conversión de la función `CONVERT` con tres argumentos de fecha y hora.
Azure Synapse Analytics	Amazon Redshift	Se mejoró la conversión de la función `DATEDIFF`.
Azure Synapse Analytics Microsoft SQL Server DW	Amazon Redshift	Se actualizó la versión del paquete de extensión. Asegúrese de aplicar la última versión del paquete de extensión a sus AWS SCT proyectos existentes. Para obtener más información, consulte Uso de paquetes de extensión.
BigQuery	Amazon Redshift	Se resolvió un problema que impedía la conversión de los objetos filtrados en el modo de interfaz de la línea de comandos (CLI).
Greenplum	Amazon Redshift	Se ha corregido un error AWS SCT que impedía convertir las tablas temporales declaradas en un procedimiento almacenado.
Greenplum	Amazon Redshift	Se corrigió un error que provocaba la omisión de atributos de codificación de columnas en el código convertido.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de instrucciones `UPDATE` para tablas con autorreferencia que tienen más de una cláusula `INNER JOIN`.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se implementó compatibilidad de tablas temporales `inserted` y `deleted` que SQL Server utiliza para los desencadenadores de DML.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de tipos definidos por el usuario en procedimientos almacenados que se crearon en diferentes esquemas de bases de datos. Se resolvió un problema por el que no se AWS SCT podía encontrar el tipo de datos y se mostraba el elemento de acción 9996.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la aparición inesperada de corchetes (`[ ]`) alrededor de los nombres de objeto de la base de datos en el código convertido.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de funciones `@@ROWCOUNT` .
Microsoft SQL Server DW	Amazon Redshift	Se implementó la compatibilidad de los tipos de datos `geometry` y `geography` .
Microsoft SQL Server DW	Amazon Redshift	Se implementó la compatibilidad de la palabra clave `MAX` en las declaraciones de tipos de datos del código convertido.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server DW	Amazon Redshift	Se mejoró la conversión de funciones `DATEADD`.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de código SQL en aplicaciones Java al agregar soporte para el MyBatis marco. Para obtener más información, consulte Convertir el código SQL en aplicaciones de Java.
Oracle	Aurora PostgreSQL PostgreSQL	Se ha mejorado la conversión de código SQL en las aplicaciones Java que utilizan el MyBatis marco. Se agregó un elemento de acción 30411 para el código SQL con una sintaxis no compatible.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de código SQL en las aplicaciones de Pro*C al agregar compatibilidad para declaraciones `typedef struct`.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la compatibilidad de instrucciones `CROSS JOIN` y `LEFT JOIN`.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de instrucciones `MERGE`. Se resolvió un problema que provocaba la omisión de valores que se iban a insertar en el código convertido.
Teradata	Amazon Redshift	Se modificó la configuración de codificación de compresión de columnas predeterminada que AWS SCT utiliza en el código convertido para que coincida con la configuración predeterminada de Amazon Redshift. Para obtener más información, consulte Codificaciones de compresión en la Guía para desarrolladores de bases de datos de Amazon Redshift.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de las operaciones matemáticas que utilizan el tipo de datos `TIME`.
Teradata	RSQL de Amazon Redshift	Se implementó la conversión del FastExport código que se encuentra dentro de los scripts de shell.
BTEQ de Teradata	RSQL de Amazon Redshift	Se ha corregido un error por el AWS SCT que no se convertían `%data` las sentencias `COALESCE` AND.
Vertica	Amazon Redshift	Se mejoraron las sugerencias de optimización de la conversión cuando un usuario selecciona una estrategia de optimización.

Notas de publicación de la AWS SCT compilación 666

Origen	Destino	Novedades, mejoras o correcciones
Base de datos SQL Azure	Aurora PostgreSQL	Se resolvió un error de análisis que se producía en cláusulas `ON` que están dentro de instrucciones `JOIN`.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	PostgreSQL	
Azure Synapse Analytics	Amazon Redshift	Se agregaron tres funciones nuevas al paquete de extensión para mejorar la conversión de la función `CONVERT` con argumentos de fecha y hora.
Azure Synapse Analytics	Amazon Redshift	Se mejoró el cargador de metadatos de origen para garantizar que AWS SCT cargue los esquemas de bases de datos del sistema.
Azure Synapse Analytics	Amazon Redshift	Se corrigió un error del solucionador que se producía en las columnas de tablas temporales.
Azure Synapse Analytics	Amazon Redshift	Se implementó la conversión de tipos de datos `BINARY` y `VARBINARY` al tipo de datos `VARBYTE`.
Azure Synapse Analytics	Amazon Redshift	Se implementó la compatibilidad del tipo de datos `TIME` en el código convertido.
Azure Synapse Analytics	Amazon Redshift	Se mejoró la conversión de cláusulas `COLLATE`. Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 31141 durante la conversión de columnas con la intercalación de bases de datos predeterminadas.
BigQuery	Amazon Redshift	Se implementó la conversión de procedimientos que cambian los parámetros de entrada.
Greenplum	Amazon Redshift	Se ha resuelto un problema que provocaba que AWS SCT se utilizara una consulta que no era compatible con las bases de datos de Greenplum 6.x.

Origen	Destino	Novedades, mejoras o correcciones
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de secciones `EXCEPTION` mediante la transferencia de controladores de excepciones de Db2 para z/OS a PostgreSQL.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de instrucciones `OPEN CURSOR`.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de funciones `IIF` mediante el uso de expresiones `CASE`.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de procedimientos con parámetros con valores de tabla cuando la instrucción `CREATE PROCEDURE` no incluía un bloque `BEGIN...END` .
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de la función `SCOPE_IDENTITY` .

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Amazon RDS para Oracle	Se corrigió un error del cargador que se producía con el rol `SELECT_CATALOG_ROLE` cuando se utilizaba Oracle 10g como origen.
Oracle	Amazon RDS para Oracle	Se mejoró el cargador para que sea compatible con los trabajos del Programador de Oracle.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de instrucciones `JOIN` con cláusulas `USING`.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró el rendimiento del código convertido cuando el código fuente incluye variables globales en cláusulas `WHERE`.
Oracle	Aurora PostgreSQL PostgreSQL	Se ha mejorado la conversión de código SQL en aplicaciones Java al añadir soporte para el MyBatis marco. Para obtener más información, consulte Convertir el código SQL en aplicaciones de Java.
Oracle DW	Amazon Redshift	Se implementó la conversión de los operadores relacionales `PIVOT` y `UNPIVOT`.
Teradata	Amazon Redshift	Se corrigió un error que impedía la conversión de código fuente que usa objetos JSON.
Teradata	Amazon Redshift	Se corrigió un error que impedía la carga correcta de las tablas creadas por un usuario eliminado.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se implementó la conversión de funciones `INSTR` en funciones `STRPOS` nativas de Amazon Redshift.
Teradata	Amazon Redshift	Se implementó la conversión de las funciones `NVP` y `TRANSLATE` .
Teradata	Amazon Redshift	Se mejoró la conversión de expresiones `COALESCE`.
Teradata	Amazon Redshift	Se mejoró la conversión de instrucciones `DECLARE CONDITION` .
Teradata	Amazon Redshift	Se mejoró la conversión de funciones `EXTRACT` con el elemento de sintaxis `SECOND`.
Teradata	Amazon Redshift	Se mejoró la conversión de las variables `SQLSTATE` y `SQLCODE` y de las instrucciones `LOOP`.
Teradata	Amazon Redshift	Se mejoró la conversión de índices únicos.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9996 durante la conversión de instrucciones `CURRENT_TIMESTAMP` con la precisión fraccional establecida en 3.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de las barras invertidas en cadenas literales.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la inclusión de un nombre de campo incorrecto en la instrucción `ADD CONSTRAINT` en instrucciones `EXEC` convertidas.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la inclusión de un nombre de subconsulta incorrecto en las subconsultas `QUALIFY` convertidas.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se resolvió un problema que impedía la aplicación de vistas convertidas. Se agregó una conversión explícita a un tipo de datos específico para los valores NULL en el código convertido.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de funciones de fecha y hora.
Teradata	Amazon Redshift	Se resolvió un problema que impedía la conversión de literales de cadena hexadecimales.

Notas de publicación de la AWS SCT compilación 665

Origen	Destino	Novedades, mejoras o correcciones
Azure Synapse Analytics	Amazon Redshift	Se implementó la conversión de funciones `CONCAT` con argumentos `VARCHAR`.
Azure Synapse Analytics	Amazon Redshift	Se ha mejorado la conversión de `CREATE TABLE` sentencias que crean tablas temporales y no incluyen el nombre del esquema. AWS SCT crea el dbo esquema para almacenar estas tablas temporales en la base de datos de destino.
Azure Synapse Analytics	Amazon Redshift	Se mejoró la conversión de instrucciones `DROP TABLE` que se ejecutan en tablas temporales.
Azure Synapse Analytics	Amazon Redshift	Se mejoró la conversión de instrucciones `OBJECT_ID` con los bloques `BEGIN...END` .
Azure Synapse Analytics	Amazon Redshift	Se resolvió un error que AWS SCT impedía convertir los procedimi entos almacenados con comentarios de bloque.

Origen	Destino	Novedades, mejoras o correcciones
BigQuery	Amazon Redshift	Se implementó la conversión de almacenes de BigQuery datos a Amazon Redshift. Para obtener más información, consulte Usar BigQuery como origen para AWS SCT.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de desencadenadores que gestionan varios eventos y funcionan con tablas de sistema `inserted` y `deleted` en SQL Server.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se corrigió un error del solucionador que se producía en las tablas del sistema `inserted` y `deleted` en SQL Server.
Microsoft SQL Server	Babelfish para Aurora PostgreSQL	Se implementó la compatibilidad con la nueva versión 2.1.0 del archivo de configuración de características de Babelfish. En este archivo se definen las características de SQL que son compatibles y no compatibles con versiones específicas de Babelfish.
Oracle	Aurora MySQL MariaDB MySQL	Se resolvió un problema que provocaba la conversión incorrecta del tipo de datos `varchar2`.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora MySQL Aurora PostgreSQL MariaDB MySQL PostgreSQL	Para las bases de datos Oracle de la versión 12c y versiones posteriores, AWS SCT admite los siguientes tipos de datos ampliados: • `VARCHAR2` • `NVARCHAR2` • `RAW` AWS SCT aumentó la longitud máxima de columna admitida de 8.000 a 32.767 bytes para estos tipos de datos.
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un error de análisis que se producía en el paquete Oracle Event Processing.
Teradata	Amazon Redshift	Se agregó un elemento de acción 13214 para varias cláusulas `RESET WHEN` en una única declaración `SELECT`.
Teradata	Amazon Redshift	Se agregó un elemento de acción para variables `SQLSTATE` que se encuentran fuera de un bloque de gestión de excepciones.
Teradata	Amazon Redshift	Se implementó la conversión de variables `ACTIVITY_COUNT` a `ROW_COUNT` .
Teradata	Amazon Redshift	Se implementó la conversión de la función `ST_TRANSFORM` de geometría integrada.
Teradata	Amazon Redshift	Se mejoró la conversión de instrucciones eliminadas en vista sin la cláusula `WHERE`.
Teradata	Amazon Redshift	Se mejoró la conversión de operadores `CAST` en expresiones.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se mejoró la conversión de cláusulas `GROUP BY`.
Teradata	Amazon Redshift	Se mejoró la conversión de las funciones `INSTR` y `REGEXP_INSTR` integradas.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de referencias a los alias de las columnas laterales.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de los nombres de las columnas en la subconsulta `QUALIFY`.
Teradata	Amazon Redshift	Se implementó la conversión de comandos `.QUIT` con la palabra clave de valor de estado `ERRORCODE` .
BTEQ de Teradata	RSQL de Amazon Redshift	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9996 durante la conversión de instrucciones `CREATE`.
BTEQ de Teradata	RSQL de Amazon Redshift	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9998 durante la conversión de instrucciones `END`.

Notas de la versión de la compilación AWS SCT 664

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se agregó compatibilidad con Amazon Redshift sin servidor como origen y destino para los proyectos de migración de bases de datos en AWS SCT. Para conectarse a Amazon Redshift sin servidor, utilice el controlador JDBC versión 2.1.0.9 o posterior de Amazon Redshift.
Todos	Todos	Se ha mejorado la interfaz de usuario de la ventana de configuración de conversión. AWS SCT ahora muestra la configuración solo para

Origen	Destino	Novedades, mejoras o correcciones
		los pares de conversión de bases de datos con reglas de mapeo creadas. Para obtener más información, consulte Creación de reglas de asignación.
Todos	Todos	Se actualizó el informe de evaluación para eliminar la información duplicada sobre la línea y la posición del elemento de acción.
Todos	Amazon Redshift	Se implementó el equilibrado automático de memoria en las tareas de extracción de datos.
Todos	Amazon Redshift	Se resolvió un error que impedía a los agentes de extracción de datos conectarse a dispositivos AWS Snowball .
Base de datos SQL Azure IBM DB2 para z/OS IBM Db2 LUW Microsoft SQL Server MySQL Oracle PostgreSQL SAP ASE	Aurora MySQL Aurora PostgreSQL MariaDB MySQL PostgreSQL	Se implementó la compatibilidad con SUSE Linux 15.3 como plataforma para ejecutar agentes de extracción de datos.

Origen	Destino	Novedades, mejoras o correcciones
Azure Synapse Analytics	Amazon Redshift	Se mejoró la conversión de funciones `DATEADD`.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se agregó la capacidad de cambiar la intercalación de columnas en las reglas de migración.
Microsoft SSIS	AWS Glue AWS Glue Studio	Se resolvió un error inesperado que se producía cuando los usuarios seleccionaban un script de origen.
Oracle	Aurora MySQL MariaDB MySQL	Se implementó la conversión del uso de funciones almacenadas como expresiones de columnas generadas. AWS SCT crea activadores para emular este comportamiento porque MySQL no admite el uso de funciones almacenadas como expresiones de columnas generadas.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de funciones del `UTL_MATCH` paquete como parte del paquete de AWS SCT extensiones.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de la función `REGEXP_LIKE` con el parámetro `NULL`.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de la función `SYS_EXTRACT_UTC` .
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de código SQL en aplicaciones de C++ + mediante la implementación de compatibilidad para funciones `Wcscats`, `Wcscpys` y `Wcsncats`. Para obtener más información, consulte Convertir el código SQL de aplicaciones de C++ con AWS SCT.
Oracle DW Snowflake	Amazon Redshift	Se resolvió un problema que impedía que las declaraciones convertidas incluyeran la conversión explícita de valores al tipo de datos de la columna. Este problema se producía en las instrucciones que utilizaban los resultados de consultas de otras tablas.
Teradata	Amazon Redshift	Se agregó la capacidad de cambiar la intercalación de columnas entre `case sensitive` y `case insensitive` en reglas de migración. Para obtener más información, consulte Crear reglas de migración.
Teradata	Amazon Redshift	Se corrigió un error del solucionador que se producía en instrucciones `CREATE TABLE AS` .
Teradata	Amazon Redshift	Se corrigió un error que impedía la conversión de la función `P_INTERSECT` integrada con una expresión `COALESCE`.
Teradata	Amazon Redshift	Se implementó la conversión de columnas denominadas `OID` a `_OID` para evitar el uso de una palabra clave reservada en Amazon Redshift.
Teradata	Amazon Redshift	Se implementó la conversión de instrucciones `RENAME` para funciones, procedimientos, vistas y macros.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se implementó la conversión de las funciones `STROKE` en la función `SPLIT_PART` de Amazon Redshift.
Teradata	Amazon Redshift	Se mejoró la conversión de las funciones del sistema `INSTR` y `REGEXP_INSTR` .
Teradata	Amazon Redshift	Se mejoró la conversión del tipo de datos `TIME`.
Teradata	Amazon Redshift	Se mejoró la simulación de las tablas `SET` y `MULTISET` mediante la implementación de la conversión de índices únicos principales y secundarios.
Teradata	Amazon Redshift	Se resolvió un error de análisis que se producía en la función `CHARACTER` .
BTEQ de Teradata	RSQL de Amazon Redshift	Se ha resuelto un error que se producía cuando los usuarios eliminaban del proyecto los scripts de Teradata Basic Teradata Query (BTEQ) de Teradata. AWS SCT

AWS SCT Notas de publicación de la compilación 663

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se agregó la posibilidad de cambiar la longitud de los tipos de datos `char`, `varchar`, `nvarchar` y `string` mediante el operador de multiplicación en una regla de migración. Para obtener más informaci ón, consulte Crear reglas de migración.
Todos	Todos	Se implementó la compatibilidad con tres columnas nuevas en el informe de evaluación multiservidor y se actualizó el formato del archivo de entrada. Utilice la plantilla actualizada del archivo de entrada con la versión más reciente de AWS SCT. Para obtener más información, consulte Crear un informe de evaluación multiservidor para la migración de la base de datos.

Origen	Destino	Novedades, mejoras o correcciones
Azure Synapse Analytics	Amazon Redshift	Se mejoró la conversión de instrucciones `OBJECT_ID` .
Microsoft SQL Server	Babelfish para Aurora PostgreSQL	Se agregó compatibilidad con Babelfish para Aurora PostgreSQL 1.2.0 como plataforma de destino para los informes de evaluación de la migración de bases de datos. Para obtener más información, consulte Funcionalidad compatible en Babelfish por versión en la Guía del usuario de Amazon Aurora.
Microsoft SQL Server DW	Amazon Redshift	Se ha agregado compatibilidad para cláusulas `AT TIME ZONE`.
Microsoft SQL Server DW	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de una instrucción fuera del bloque `BEGIN/END` .
Netezza	Amazon Redshift	Se mejoró la conversión del tipo de datos `TIME` y se implementó la conversión de funciones integradas, expresiones y literales relaciona dos.
Oracle	Aurora PostgreSQL PostgreSQL	Se corrigió un error del cargador que se producía al usar Oracle 10g como origen.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de cláusulas `OFFSET` y `FETCH`.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de procedimientos con parámetros `OUT` con valores predeterminados.
Oracle DW	Amazon Redshift	Se mejoró la conversión de funciones de Oracle en funciones definidas por el usuario de Amazon Redshift.
Snowflake	Amazon Redshift	Se mejoró la conversión de cláusulas `WITH`.
Teradata	Amazon Redshift	Se agregó un nuevo elemento de acción 13209 para los caracteres multibyte no compatibles para el tipo de datos `CHAR`.
Teradata	Amazon Redshift	Se corrigió un error del cargador que impedía la carga completa de las tablas.
Teradata	Amazon Redshift	Se corrigió un error del transformador que impedía la conversión de la función `P_INTERSECT` integrada en una condición `JOIN`.
Teradata	Amazon Redshift	Se corrigió un error que provocaba que el nombre de una vista se convirtiera en mayúsculas y minúsculas de forma errónea cuando la instrucción `SELECT` se ejecutaba en una tabla con caracteres especiales en el nombre.
Teradata	Amazon Redshift	Se mejoró la conversión de instrucciones `INSERT` con el valor `UNTIL_CHANGED` en el tipo de datos `PERIOD(DATE)` .
Teradata	Amazon Redshift	Se mejoró la conversión de la función `FORMAT` integrada mediante la función `TO_CHAR` de Amazon Redshift.
Teradata	Amazon Redshift	Se mejoró la conversión de la función `RANK` integrada para garantizar que el código convertido devuelva valores NULL en el mismo orden que el código fuente.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se mejoró la conversión de restricciones únicas, como los índices únicos principales o secundarios.

Notas de publicación de la AWS SCT compilación 662

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se agregó la capacidad de crear automáticamente AWS SCT proyectos para cada base de datos fuente al crear el informe de evaluación multiservidor. Con esta opción activada, AWS SCT puede añadir reglas de mapeo a estos proyectos y guardar las estadísticas de conversión para utilizarlas sin conexión a Internet. Para obtener más información, consulte Crear un informe de evaluación multiservidor para la migración de la base de datos.
Todos	Todos	Se implementó la compatibilidad del porcentaje (%) como carácter comodín en los nombres de bases de datos y esquemas al crear el informe de evaluación multiservidor.
Todos	Aurora MySQL Aurora PostgreSQL	Se actualizó el tiempo de ejecución de todas AWS Lambda las funciones a la versión 3.9 de Python.
Todos	Amazon Redshift	Se actualizaron todos los agentes de extracción de datos para su uso AWS SDK for Java 2.x.
Base de datos SQL Azure	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de instrucciones `DELETE` con cláusulas `NOT EXISTS` .

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server		
Azure Synapse Analytics	Amazon Redshift	Se resolvió un error que provocaba un fallo en la conexión a una base de datos de origen.
IBM DB2 para z/OS	Aurora PostgreSQL PostgreSQL	Se resolvió un error por el que el código convertido de un desencadenador incluía dos menciones del alias del objeto.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de objetos con nombres en mayúsculas y minúsculas cuando la opción Distinguir entre mayúsculas y minúsculas en los nombres de los objetos de la base de datos está activada.
Microsoft SQL Server DW Teradata	Amazon Redshift	Se implementó la conversión de los operadores relacionales `PIVOT` y `UNPIVOT`.
Netezza	Amazon Redshift	Se implementó la conversión del tipo de datos `TIME`.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Se implementó la conversión constante del paquete `UTL_TCP.CRLF` .
Oracle	Aurora PostgreSQL PostgreSQL	Se corrigió un problema del paquete de extensión que impedía mantener la longitud de los tipos de datos de las columnas de longitud variable durante la conversión.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la conversión de código SQL en aplicaciones de C++. Para obtener más información, consulte Convertir el código SQL de aplicaciones de C++ con AWS SCT.
Oracle	Aurora PostgreSQL PostgreSQL	Se implementó la compatibilidad con la nomenclatura que distingue mayúsculas y minúsculas para la conversión de variables globales y matrices asociativas.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de las funciones `TO_CHAR`, `TO_DATE` y `TO_NUMBER` del paquete de extensión.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión del operador `TABLE()`.
Oracle DW	Amazon Redshift	Se agregó compatibilidad para la conversión de claves principales y otras restricciones.
Oracle DW	Amazon Redshift	Se corrigió un problema que impedía la aparición del elemento de acción 12054 durante la conversión de instrucciones condicionales.
SAP ASE	Aurora PostgreSQL PostgreSQL	Se resolvió un error que se producía cuando se creaba un objeto con un nombre vacío en el árbol de destino durante la conversión de tablas con columnas del tipo definido por el usuario.
SAP ASE	Aurora PostgreSQL PostgreSQL	Se corrigió un error del cargador para los objetos almacenados, como scripts, rutinas, etc.
Snowflake	Amazon Redshift	Se ha corregido un problema por el que el elemento de acción 22152 no aparecía cuando era necesario y AWS SCT mostraba el resultado de la conversión como un comentario.

Origen	Destino	Novedades, mejoras o correcciones
Snowflake	Amazon Redshift	Se mejoró la conversión de las funciones de fecha y hora; se implementó la compatibilidad para las zonas horarias.
Snowflake	Amazon Redshift	Se resolvió un problema que provocaba la conversión de las expresiones de tabla comunes (CTE) no recursivas con una cláusula `WITH` en CTE recursivas.
Teradata	Amazon Redshift	Se mejoró la conversión de instrucciones `UPDATE` con enlaces a tablas en buen estado.
Teradata	Amazon Redshift	Se mejoró la conversión de instrucciones `RENAME TABLE`.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la aparición de columnas vacías en el archivo de valores separados por comas (CSV) con un informe de evaluación.
Teradata	RSQL de Amazon Redshift	Se corrigió un error que provocaba la omisión de un punto y coma al final de una macro de Basic Teradata Query (BTEQ).
Teradata	RSQL de Amazon Redshift	Se mejoró la conversión de valores de tipos de datos múltiples en instrucciones `CASE`.
Teradata	RSQL de Amazon Redshift	Se mejoró la conversión de la cláusula `LIKE ANY` con un carácter `ESCAPE`.
Teradata	RSQL de Amazon Redshift	Se mejoró la conversión de la función `CAST` en instrucciones `INSERT`.
Teradata	RSQL de Amazon Redshift	Se mejoró la conversión de las zonas horarias, se implementó la asignación de las regiones de las zonas horarias.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	RSQL de Amazon Redshift	Se resolvió un problema que provocaba la aparición inesperada del elemento de acción 9998 durante la conversión de scripts del intérprete de comandos con scripts de BTEQ.
Teradata	RSQL de Amazon Redshift AWS Glue	Se implementó el límite de 500 caracteres para los valores de las variables de sustitución.
Vertica	Amazon Redshift	Se implementó la conversión de tipos de datos `BINARY`, `VARBINARY` , `LONG BINARY`, `BYTEA`y `RAW` al tipo de datos `VARBYTE`.
Vertica	Amazon Redshift	Se mejoró la conversión de los literales y las funciones integradas.

Notas de la versión de la AWS SCT compilación 661

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se agregaron filtros para buscar reglas de asignación en la vista de asignación. Al aplicar un filtro, AWS SCT muestra las reglas que coinciden con las condiciones de filtrado de la lista de mapeos del servidor. Para obtener más información, consulte Administración de reglas de asignación.
Todos	Todos	Se actualizó Apache Log4j a la versión 2.17.1.
Todos	Amazon Redshift	Se agregó compatibilidad con la migración de datos a Amazon Redshift mediante la cláusula `ENCRYPTED` del comando `COPY`.
Todos	Amazon Redshift	Se mejoró la API de REST de los agentes de extracción de datos. La API de REST actualizada añade compatibilidad con nuevas propiedades, como la clave de cifrado, el tipo de cifrado, etc.

Origen	Destino	Novedades, mejoras o correcciones
Todos	Amazon Redshift	Se implementó la asunción de roles en los agentes de extracción de datos. Esta actualización mejora la distribución de las tareas secundarias y permite a AWS SCT asignar tareas a los agentes libres del rol especificado.
Todos	Amazon Redshift	Se implementó una comprobación de que todos los componentes necesarios estén instalados antes de aplicar el paquete de extensión a Amazon Redshift.
Azure Synapse Analytics Microsoft SQL Server DW	Amazon Redshift	Se mejoró la conversión de las funciones del sistema `ERROR_LINE` , `ERROR_MESSAGE` , `ERROR_NUMBER` , `ERROR_PROCEDURE` , `ERROR_SEVERITY` y `ERROR_STATE` para la gestión de errores.
IBM DB2 para z/OS	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Se agregó compatibilidad para IBM DB2 para z/OS versión 12 como origen para proyectos de migración de bases de datos en AWS SCT. Para obtener más información, consulte Usar IBM Db2 para z/OS como origen.
IBM Db2 LUW	Todos	Se ha mejorado el cargador de metadatos de origen para garantizar que AWS SCT carga los parámetros rutinarios que duplican los nombres de las columnas.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se corrigió un error del transformador en los procedimientos con la instrucción `SET NOCOUNT ON`.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de la función `CONCAT` cuando un valor de entrada es una variable del tipo definido por el usuario.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de la función `DATEPART`.
Microsoft SQL Server	Babelfish para Aurora PostgreSQL	Se implementó compatibilidad para la nueva versión del archivo de configuración de características de Babelfish. En este archivo se definen las características de SQL que son compatibles y no compatibles con versiones específicas de Babelfish.
Microsoft SQL Server DW	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de una instrucción `EXECUTE`.
Microsoft SSIS	AWS Glue	Se ha mejorado la interfaz de usuario del asistente de configuración de tareas. AWS SCT ahora muestra solo las conexiones disponibles en la sección de configuración de conexiones.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SSIS	AWS Glue	Se resolvió un problema que impedía la aplicación de las reglas de transformación a las tareas de paquetes y a las reglas variables.
Microsoft SSIS	AWS Glue AWS Glue Studio	Se agregó un nuevo elemento de acción 25042 para los component es no compatibles.
Microsoft SSIS	AWS Glue Studio	Se implementó la conversión de los paquetes de extracción, transformación y carga (ETL) de Microsoft SQL Server Integration Services (SSIS) a AWS Glue Studio. Para obtener más información, consulte Convertir SSIS a AWS Glue Studio.
Oracle	MariaDB	Se corrigió un problema con la conversión del operador `MINUS`.
Oracle	MariaDB	Se mejoró la conversión de las funciones `ROWNUM`, `SYS_GUID`, `TO_CHAR` y `ADD_MONTHS` , y cuando la variable del sistema `sql_mode` de MariaDB es Oracle.
Oracle	PostgreSQL	Se agregó una opción para evitar la conversión de tipos de variables de enlace a tipos de SQL en proyectos de conversión de aplicacio nes genéricas.
Oracle	PostgreSQL	Se agregó una opción para evitar agregar el nombre del esquema al nombre del objeto convertido en los proyectos de conversión de aplicaciones genéricas.
Oracle	PostgreSQL	Se añadió compatibilidad con el formato de variables de enlace `?x` para la conversión de código SQL de aplicaciones.
Oracle DW	Amazon Redshift	Se implementó la conversión del tipo de datos `RAW` al tipo de datos `VARBYTE`.
Teradata	Amazon Redshift	Se agregó una opción para simular tablas `SET` en el código convertid o. Para esta emulación, AWS SCT soportes `MIN` y `MAX` condiciones.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se mejoró la conversión de las operaciones de unión que tienen parámetros de distintos tipos de datos. Esta actualización permite AWS SCT aplicar reglas de transformación durante la conversión de dichas operaciones.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de la cláusula `GROUP BY`.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de la cláusula `QUALIFY`.
Teradata	Amazon Redshift	Se ha resuelto un error inesperado que se producía durante la importación de FastExport los scripts.
Teradata	RSQL de Amazon Redshift	Se implementó la capacidad de editar los valores de las variables en los scripts del intérprete de comandos y de BTEQ de Teradata.
Teradata	RSQL de Amazon Redshift	Se resolvió un problema por el que faltaba el script de manifiesto para las sesiones convertidas de Teradata FastLoad .
Teradata	RSQL de Amazon Redshift	Se resolvió un problema por el que faltaba la extensión del archivo de manifiesto en el localizador uniforme de recursos (URL) de los scripts convertidos. FastLoad
BTEQ de Teradata	RSQL de Amazon Redshift	Se corrigió un error del cargador para los scripts con variables de sustitución.
BTEQ de Teradata	RSQL de Amazon Redshift	Se corrigió un problema que impedía la aparición del elemento de acción 27022 cuando era necesario.

Notas de la versión de la AWS SCT compilación 660

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se agregó la AWS Secrets Manager compatibilidad con Secure Sockets Layer (SSL) en el informe de evaluación multiservidor. Para obtener más información, consulte Crear un informe de evaluación multiservidor para la migración de la base de datos.
Todos	Todos	Se mejoró la recopilación de estadísticas para los objetos convertidos.
Todos	PostgreSQL	Se implementó la compatibilidad con la versión principal 14 de PostgreSQL y MariaDB 10.6 como destinos de migración.
Azure Synapse Analytics	Amazon Redshift	Se mejoró la lógica de transformación para los nombres de los objetos convertidos.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL	Se mejoró la conversión del tipo de datos `XML`.
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que provocaba la conversión incorrecta de cláusulas `NOT LIKE`.
Microsoft Azure SQL Database	Aurora PostgreSQL	Se corrigió un error del transformador en los procedimientos con instrucciones `INSERT`, `DELETE` y `UPDATE` que incluyen la cláusula `OUTPUT`.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	PostgreSQL	
Microsoft Azure SQL Database Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se corrigió un error del transformador en los procedimientos con la instrucción `RETURN @@ROWCOUNT` .
Microsoft SQL Server	Todos	Se mejoró la conversión de los procedimientos que utilizan servidores enlazados.
Microsoft SQL Server	Todos	Se agregó compatibilidad para la autenticación de Microsoft Windows en el informe de evaluación multiservidor.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se corrigió un error del transformador en los constructores de valores de tabla.
Microsoft SQL Server DW	Amazon Redshift y AWS Glue	Compatibilidad con la conversión de scripts de extracción, transformación y carga (ETL) para incluir la ruta correcta a los scripts convertidos.
Microsoft SQL Server DW	Amazon Redshift	Se resolvió un problema que provocaba la generación de diferentes scripts convertidos para plataformas de bases de datos de destino virtuales y reales.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	PostgreSQL Aurora PostgreSQL	Se agregó compatibilidad para la conversión de índices para vistas materializadas.
Oracle	PostgreSQL Aurora PostgreSQL	Se corrigió un problema que impedía la aparición del elemento de acción 5982 al convertir las restricciones `PRIMARY KEY` y `UNIQUE` con la opción `NOVALIDATE` .
Oracle DW	Amazon Redshift	Se resolvió un problema que provocaba la visualización de categoría s adicionales en el esquema convertido.
Teradata	Amazon Redshift	Se corrigió un problema que impedía la aparición del elemento de acción 13185 al convertir una columna sin resolver como argumento de la función `CAST`.
Teradata	Amazon Redshift	Se mejoró la conversión de instrucciones `DELETE` y `DELETE ALL` para usar el comando `TRUNCATE` en el código convertido.
Teradata	Amazon Redshift	Se mejoró la conversión de tablas `SET`.
Teradata	Amazon Redshift	Se mejoró la conversión mejorada de la condición `NORMALIZE` .
Teradata	Amazon Redshift	Se actualizó el informe de evaluación para eliminar las estadísticas de conversión del esquema de la base de datos de la lista de objetos de almacenamiento de la base de datos.
Teradata	Amazon Redshift	Se mejoró la conversión de la instrucción `UPDATE` sin la cláusula `FROM`.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se implementó la compatibilidad del tipo de datos `VARBYTE` en el código convertido.
BTEQ de Teradata	AWS Glue	Se resolvió un problema que provocaba la desactivación de la opción Convertir AWS Glue en el menú contextual.
BTEQ de Teradata	RSQL de Amazon Redshift	Se resolvió un problema que provocaba la omisión de tipos de datos en el código convertido.
BTEQ de Teradata	RSQL de Amazon Redshift	Se resolvió un problema que provocaba la aparición incorrecta de variables de sustitución entre comillas en el código convertido.
BTEQ de Teradata	RSQL de Amazon Redshift	Se ha corregido un problema relacionado con la conversión de variables de sustitución con valores en los FastLoad scripts.
Vertica	Amazon Redshift	Se implementó la compatibilidad del tipo de datos `TIME` en el código convertido.
Vertica	Amazon Redshift	Se mejoró la conversión de las expresiones `SELECT DISTINCT` y `ORDER BY`.
Vertica	Amazon Redshift	Se agregó compatibilidad para la conversión restricciones.
Vertica	Amazon Redshift	Se corrigió un error que impedía que el informe de evaluación se guardara como archivo CSV (valores separados por comas).

Notas de publicación de la AWS SCT versión 659

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se mejoró el Asistente de proyecto nuevo, que genera un informe de evaluación combinado para varias bases de datos de origen.
Todos	Todos	Se corrigió un problema que impedía la creación del paquete de extensión en proyectos que incluían varias bases de datos de origen y destino.
Todos	Todos	Se mejoró la conversión del código SQL que está incrustado en el código fuente de la aplicación.
Todos	Todos	Se agregó la capacidad de ejecutar scripts desde diferentes carpetas en la interfaz de AWS SCT línea de comandos.
Todos	Amazon Redshift	Se mejoró el mensaje de advertencia que aparece cuando los usuarios eligen la opción Ejecutar la optimización en proyectos de migración con la plataforma de base de datos de destino virtual Amazon Redshift.
Todos	Aurora PostgreSQL	Se implementó la compatibilidad con la versión principal 13 de PostgreSQL en Aurora PostgreSQL-Compatible Edition como destino de migración.
Todos	Amazon RDS para MySQL	Se implementó de forma predeterminada la conversión de código que no distingue entre mayúsculas y minúsculas.
Azure Synapse Analytics	Amazon Redshift	Se resolvió un error que provocaba el fallo de la conexión a una base de datos de origen en la interfaz de la línea de comandos.
Microsoft SQL Server	PostgreSQL	Se mejoró la conversión de procedimientos que incluyen instrucciones `UPDATE` con condiciones de unión.

Origen	Destino	Novedades, mejoras o correcciones
	Aurora PostgreSQL	
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Se mejoró la conversión de los desencadenadores, los procedimientos almacenados y las funciones que incluyen el valor después del signo igual.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Se corrigió un error del transformador en los procedimientos con la instrucción `DELETE` y el operador `OR`.
Microsoft SQL Server	PostgreSQL Aurora PostgreSQL	Se mejoró la conversión de la cláusula `OUTPUT`.
Microsoft SQL Server DW	Amazon Redshift y AWS Glue	Se mejoró la conversión del tipo de datos `NUMERIC`.
Microsoft SQL Server DW	Amazon Redshift	Se mejoró la conversión de las vistas que tienen un alias de tabla con el mismo nombre que la tabla original.
Microsoft SSIS	AWS Glue	Se solucionó un problema por el que las credenciales de AWS Glue conexión no se mostraban en la ventana Configurar conexiones.

Origen	Destino	Novedades, mejoras o correcciones
Netezza	Amazon Redshift	Se agregó la posibilidad de repetir la ejecución de tareas de migración de datos con captura de datos de cambios (CDC) todos los días.
Netezza	Amazon Redshift	Se corrigió un error que provocaba que la pestaña Tareas quedara inactiva tras anular el registro de un extractor de datos.
Netezza	Amazon Redshift	Se corrigió un problema que impedía la aparición de la confirmac ión del registro del agente de migración de datos en la interfaz de usuario.
Netezza	Amazon Redshift	Se corrigió un problema que provocaba el fallo de una base de datos de origen con un Error de carga.
Netezza	Amazon Redshift	Se resolvió un error que impedía la ejecución de los agentes de migración de datos tras abrir un proyecto guardado.
Oracle	Amazon RDS para Oracle	Se implementó compatibilidad para Oracle Unified Auditing.
Oracle	PostgreSQL Aurora PostgreSQL	Se implementó la conversión de código SQL en aplicaciones de C#. Para obtener más información, consulte Convertir el código SQL de aplicaciones de C#.
Oracle	PostgreSQL Aurora PostgreSQL	Se implementó una nueva lógica de transformación para los nombres de objetos que distinguen mayúsculas de minúsculas para mejorar la visibilidad de los cambios en la conversión de código. AWS SCT convierte los nombres de los objetos en mayúsculas a minúsculas. También ocurre lo contrario: AWS SCT convierte los nombres de los objetos en minúsculas a mayúsculas. Los demás nombres de objetos y palabras reservadas se convierten sin cambios.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	PostgreSQL Aurora PostgreSQL	Se mejoró la conversión de particiones hash sin la restricción `NOT NULL`.
Oracle	Aurora PostgreSQL	Se añadió compatibilidad para la conversión de las restricciones `CHECK`, `FOREIGN KEY` y `NOT NULL` de Oracle con la cláusula `ENABLE NOVALIDATE` .
Oracle DW	Amazon Redshift	Se corrigió un problema que provocaba la migración de valores incorrectos de los números de punto flotante.
Oracle DW	Amazon Redshift y AWS Glue	Se resolvió un problema que provocaba la aparición de columnas vacías en el informe de evaluación de la migración de base de datos en un archivo de valores separados por comas (CSV).
SAP ASE	PostgreSQL Aurora PostgreSQL	Se corrigió un problema que provocaba una interrupción inesperada de la conversión.
Snowflake	Amazon Redshift	Se mejoró la conversión del tipo de datos `VARIANT`.
Teradata	Amazon Redshift	Se mejoró la conversión de la instrucción `COLLECT STATISTICS` .
Teradata	Amazon Redshift	Se corrigió un problema que impedía la aparición del elemento de acción 9998 al convertir vistas anidadas con columnas `PERIOD`.
Teradata	Amazon Redshift y AWS Glue	Se solucionó un problema por el que una plataforma de AWS Glue destino virtual no aparecía en la interfaz de usuario después de abrir un proyecto guardado.

Origen	Destino	Novedades, mejoras o correcciones
BTEQ de Teradata	AWS Glue	Se ha corregido un problema por el que no se admitía la conversión a una plataforma de AWS Glue destino virtual tras abrir un proyecto guardado.
BTEQ de Teradata	RSQL de Amazon Redshift	Se mejoró el resaltado de la sintaxis del código convertido.
BTEQ de Teradata	RSQL de Amazon Redshift	Se implementó la comprobación de los valores de los parámetros tras la carga. Los valores no compatibles aparecen resaltados en la pestaña Variables.
Vertica	Amazon Redshift	Se implementó la conversión de funciones de agregación.
Vertica	Amazon Redshift	Se implementó la conversión de las proyecciones en vistas materiali zadas y se mejoró la interfaz de usuario que muestra el código fuente de las proyecciones.

Notas de la versión de la AWS SCT compilación 658

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se proporcionó integración con AWS Secrets Manager. Ahora puede usar las credenciales de conexión a la base de datos que están almacenadas en Secrets Manager.
Todos	Todos	Se agregó soporte para scripts en formato YAML en la interfaz de línea de AWS SCT comandos.
Todos	Amazon Redshift	Se implementó compatibilidad para puntos de conexión de la interfaz Amazon S3 (VPCE) en los agentes de extracción de datos.

Origen	Destino	Novedades, mejoras o correcciones
Todos	Amazon Redshift	Se agregó compatibilidad con la plataforma de base de datos de destino virtual Amazon Redshift, además de Amazon AWS Glue Redshift y su combinación, ya compatibles.
Greenplum	Amazon Redshift	Se resolvió un problema que impedía que la opción Guardar como SQL no guardara el código SQL convertido en un archivo.
IBM Db2 LUW Microsoft Azure SQL Database Microsoft SQL Server Oracle SAP ASE	Aurora MySQL	Se mejoró la conversión para admitir características nuevas de Amazon Aurora-Compatible Edition con compatibilidad con MySQL 8.0.
Microsoft SQL Server	Aurora MySQL Aurora PostgreSQL MySQL PostgreSQL	Se corrigió un problema que impedía la aparición del elemento de acción 810 cuando era necesario.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de procedimientos con instrucciones `UPDATE`, `DELETE` y `INSERT`.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se corrigió un problema que impedía la aparición del elemento de acción 7810 cuando era necesario.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de una instrucción `EXEC` que está anidada dentro de una instrucción `IF...ELSE` .
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de vistas indexadas.
Netezza	Amazon Redshift	Se mejoraron los agentes de migración de datos mediante el seguimiento de las transacciones en vivo durante la carga completa en la operación de captura de datos de cambios (CDC). Ahora puede detener las tareas de migración de datos si está previsto que la sesión de CDC comience a una hora determinada. Además, puede ver el nivel de registro de errores en la consola después de detener una tarea con CDC.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Todos	Se mejoró el cargador de tablas para garantizar que los objetos se AWS SCT carguen con opciones para compartirlos.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de la función `SYSDATE` y se agregó la posibilidad de cambiar la zona horaria en Configuración de conversión.
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que impedía la conversión de instrucciones dinámicas.
Oracle	Aurora PostgreSQL PostgreSQL	Se corrigió un problema que provocaba que el código convertido no incluyera los nombres generados por el sistema.
Oracle Oracle DW	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de instrucciones `SELECT` anidadas dentro de los desencadenadores.
Oracle DW	Amazon Redshift	Se mejoró la conversión de las funciones `TO_DATE`, `TO_TIMESTAMP` y `TO_TIMESTAMP_TZ` del paquete de extensión.
Snowflake	Amazon Redshift	Se añadió una opción para guardar el código SQL convertido en diferentes archivos para cada objeto o cada instrucción.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se mejoró la conversión de la función `CONCAT`.
Teradata	Amazon Redshift	Se mejoró la conversión de una instrucción `SELECT` que está anidada dentro de una cláusula `WHERE`.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de las tablas `SET` y `MULTISET` después de que los usuarios descartaran y volvieran a crear una tabla.
Teradata	Amazon Redshift	Se mejoró la conversión de los procedimientos que incluyen una cláusula `WITH`.
Teradata	Amazon Redshift	Se mejoró la conversión del tipo de datos `DATE`.
Teradata	RSQL de Amazon Redshift	Se ha resuelto un problema que provocaba que se produjera un error inesperado en el transformador durante la conversión de FastExport scripts.
BTEQ de Teradata	RSQL de Amazon Redshift	Se añadió compatibilidad para la conversión de un índice de unión en una vista materializada.
BTEQ de Teradata	RSQL de Amazon Redshift	Se añadió compatibilidad para la conversión de una definición `TITLE` que incluye varias líneas.
BTEQ de Teradata	RSQL de Amazon Redshift	Se resolvió un problema que impedía la conversión del tamaño de un tipo de datos geoespaciales.
BTEQ de Teradata	RSQL de Amazon Redshift	Se corrigió un problema que provocaba la conversión de los nombres de los parámetros a minúsculas.

Origen	Destino	Novedades, mejoras o correcciones
BTEQ de Teradata	RSQL de Amazon Redshift	Se corrigió un problema que impedía la conversión de un procedimiento almacenado anidado dentro de una instrucción `MACRO`.
Vertica	Amazon Redshift	Se mejoró la conversión del operador `ALL`.
Vertica	Amazon Redshift	Se resolvió un problema que impedía la aplicación de la opción `Use Union all view?` en Configuración de conversión.
Vertica	Amazon Redshift	Se mejoró la conversión de los tipos de datos `TIME` y `TIME WITH TIMEZONE`.
Vertica	Amazon Redshift	Se resolvió un problema relacionado con la carga de tablas flexibles.

Problemas resueltos:

- Mejoras generales.

Notas de publicación de la AWS SCT compilación 657

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se actualizó Apache Log4j a la versión 2.17 para solucionar problemas de vulnerabilidad de seguridad.
Todos	Amazon Redshift	Se mejoraron los proyectos de optimización de esquemas, en los que las estadísticas de administración clave no se guardaban en el proyecto de AWS SCT .
Amazon Redshift	Amazon Redshift	Se corrigió un problema con la actualización de la información del servidor.

Origen	Destino	Novedades, mejoras o correcciones
Apache Cassandra	Amazon DynamoDB	Se ha corregido un problema con las reglas de mapeo al utilizar la interfaz de AWS SCT línea de comandos.
Apache Cassandra	Amazon DynamoDB	Se resolvió un problema que impedía la creación de la tarea de migración debido a un título actualizado en el certificado.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se corrigió un problema que impedía la aparición del elemento de acción 7672 durante la conversión de los procedimientos de Microsoft SQL Server con SQL dinámico.
Base de datos SQL Azure Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de funciones con valores de tabla.
Base de datos SQL Azure Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que impedía la conversión del argumento `OUT` de un procedimiento almacenado con el valor de retorno predeterminado en el argumento `INOUT`.
Greenplum	Amazon Redshift	Se mejoraron las estrategias de optimización mediante la búsqueda de las tablas y columnas más utilizadas de la tabla `QueryLog`.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se corrigieron los problemas relacionados con la conversión de lo siguiente: • Operador de asignación de concatenación de cadenas (+=) • Función `SCOPE_IDENTITY` • Tipo de datos `varchar(max)`
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de vistas con funciones no compatibles.
Microsoft SQL Server	Aurora PostgreSQL PostgreSQL	Se corrigió un problema que provocaba la conversión incorrecta de funciones no compatibles como argumento de otra función.
Microsoft SQL Server	Babelfish para Aurora PostgreSQL	Se mejoró la conversión de las referencias a las tablas de transición.
Microsoft SQL Server DW	Amazon Redshift	Se agregó la categoría de funciones de agregación al árbol de metadatos de la base de datos de origen.
Microsoft SQL Server DW	Amazon Redshift	Se mejoró la conversión del tipo de datos `TIME`.

Origen	Destino	Novedades, mejoras o correcciones
Azure Synapse Analytics Greenplum Netezza Microsoft SQL Server DW Snowflake Teradata	Amazon Redshift	Se corrigió un problema que impedía guardar los scripts `DROP` y `CREATE` al utilizar una plataforma de base de datos de destino virtual.
Microsoft SQL Server Integration Services	AWS Glue	Se resolvió un problema que impedía visualizar los scripts de los objetos de origen en la interfaz de usuario.
Netezza	Amazon Redshift	Se mejoraron las estrategias de optimización mediante la elección de la tabla de datos y las dimensiones adecuadas para la colocación.
Oracle	Aurora PostgreSQL PostgreSQL	Se resolvió un problema que impedía convertir correctamente los desencadenadores de Oracle, que utilizan números de secuencia.
Oracle	Aurora PostgreSQL PostgreSQL	Se mejoró la conversión de vistas con enlaces a bases de datos públicas.

Origen	Destino	Novedades, mejoras o correcciones
Oracle DW	Amazon Redshift	Se mejoraron las estrategias de optimización mediante el análisis de la cardinalidad de las columnas del índice.
Oracle DW	Amazon Redshift	Se corrigió un problema que provocaban la conversión incorrecta de las funciones escalares personalizadas definidas por el usuario con concatenación de cadenas.
Snowflake	Amazon Redshift	Se corrigió un problema que impedía la visualización de la opción Guardar como SQL en la interfaz de usuario.
Teradata	Amazon Redshift	Se corrigió un problema que provocaba el fallo de la recopilación de estadísticas con la excepción `LOADER ERROR`.
Teradata	Amazon Redshift	Se corrigió un problema que impedía la visualización de la opción Crear informe en la interfaz de usuario.
Teradata	Amazon Redshift	Se mejoró la conversión de la función `CAST`.
Teradata	Amazon Redshift	Se corrigió un error en la conversión de `ST_Line_Interpolate_Point` .
Teradata	Amazon Redshift	Se eliminó un valor inesperado de la ruta de la biblioteca python.
Teradata	RSQL de Amazon Redshift	Se ha corregido un error de resolución que aparecía durante la conversión de varios FastLoad scripts.
BTEQ de Teradata	RSQL de Amazon Redshift	Se mejoró la conversión del comando `DATABASE` y tipos de datos de geometría.
BTEQ de Teradata	AWS Glue	Se corrigió un problema que provocaba una sincronización incorrecta de los scripts de origen y destino en la interfaz de usuario.

Problemas resueltos:

- Mejoras generales.

Notas de la versión de la AWS SCT compilación 656

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se agregó compatibilidad para varias bases de datos de origen y destino en un proyecto. Los usuarios ahora pueden crear reglas de asignación para que coincidan con diferentes esquemas de bases de datos y plataformas de destino en el mismo proyecto.
Todos	Todos	Se agregó compatibilidad para plataformas de bases de datos de destino virtuales. Los usuarios ahora no necesitan conectarse a una base de datos de destino para ver cómo se AWS SCT convierte el esquema de su base de datos de origen.
Todos	Todos	Mejoras de la interfaz de usuario: • Se agregaron las opciones Conectar al servidor y Desconectar del servidor a los árboles de metadatos de origen y destino. • Se agregó una opción para eliminar un servidor de base de datos del AWS SCT proyecto.
Cassandra	Amazon DynamoDB	Se resolvió un problema de búsqueda por el que la variable `CASSANDRA_HOME` no incluía una barra inclinada (`/`) después de `cassandra.yaml` o de la carpeta `conf`.
Cassandra	Amazon DynamoDB	Se añadió compatibilidad con la imagen de máquina de Amazon (AMI) para Amazon Linux 2.
Cassandra	Amazon DynamoDB	Se mejoró el mensaje de error que se proporciona cuando se introduce una clave incorrecta para Cassandra.

Origen	Destino	Novedades, mejoras o correcciones
Cassandra	Amazon DynamoDB	Se mejoró la conversión mediante el cambio de una propiedad del archivo `cassandra-env.yaml` en función de la versión de la base de datos de destino.
Cassandra	Amazon DynamoDB	Se aumentó la versión Java del centro de datos Cassandra de destino a la versión 1.8.0.
Greenplum	Amazon Redshift	Se mejoraron las estrategias de optimización en la Configuración del proyecto.
Greenplum	Amazon Redshift	Se resolvió un problema de migración de datos que impedía la aplicación de los objetos a la base de datos con este error: `An I/O error occurred while sending to the backend .`
Greenplum Microsoft SQL Server DW	Amazon Redshift	Se resolvió un problema que impedía la visualización de la opción `Apply RTRIM to string columns` en la interfaz de usuario.
Microsoft SQL Server	Babelfish para Aurora PostgreSQL	Se agregó compatibilidad para Babelfish para Aurora PostgreSQL como plataforma de destino. Los usuarios ahora pueden crear un informe de evaluación para estimar la migración de SQL Server a Babelfish para Aurora PostgreSQL.
Netezza	Amazon Redshift	Se mejoraron las estrategias de optimización en la Configuración del proyecto.
SAP ASE	Aurora PostgreSQL PostgreSQL	Se implementó la capacidad de generar nombres únicos para los índices.

Origen	Destino	Novedades, mejoras o correcciones
SAP ASE	Aurora PostgreSQL PostgreSQL	Se corrigió un problema con una columna de índice duplicada en el script de destino.
Snowflake	Amazon Redshift	Se resolvió un problema por impedía la visualización de las opciones Ocultar esquemas vacíos, Ocultar bases de datos vacías y Ocultar bases de datos/esquemas del sistema en la interfaz de usuario.
Teradata	RSQL de Amazon Redshift	Se agregó soporte para la conversión de scripts de MultiLoad trabajo de Teradata a scripts RSQL de Amazon Redshift.
Teradata	RSQL de Amazon Redshift	Se ha corregido un problema relacionado con la conversión de variables de sustitución en scripts y scripts. FastLoad FastExport
Teradata	RSQL de Amazon Redshift	Se corrigió un problema que impedía la visualización de elementos de acción en la pestaña Elementos de acción después de cambiar de la pestaña Resumen.
Teradata	RSQL de Amazon Redshift	Se ha resuelto un problema por el que se producía un error después de generar un informe durante la conversión de FastExport scripts.
Teradata	RSQL de Amazon Redshift	Se resolvieron problemas de formato tras la conversión de scripts del intérprete de comandos.
Teradata	RSQL de Amazon Redshift	Se corrigió un problema que ahora permite agregar comentarios al elemento de acción 13177 en el script convertido.
Teradata	Amazon Redshift	Se corrigió un error en la conversión de tablas temporales.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se mejoró la conversión de la instrucción `SET QUERY_BAND` .
Teradata	Amazon Redshift	Se corrigió un error en la conversión de la operación `NORMALIZE` .
Vertica	Amazon Redshift	Se mejoró la descripción del elemento de acción 17008.

Problemas resueltos:

- Mejoras generales.

Notas de publicación de la AWS SCT compilación 655

Origen	Destino	Novedades, mejoras o correcciones
Teradata	RSQL de Amazon Redshift	Se solucionó un problema para garantizar que todos los problemas de evaluación aparecieran en los informes cuando se FastLoad utilizaban o MultiLoad se utilizaban.
Teradata	RSQL de Amazon Redshift	Se agregó soporte para la conversión de scripts de FastExport trabajo de Teradata a scripts RSQL de Amazon Redshift.
Teradata	RSQL de Amazon Redshift	Se solucionó un problema para garantizar que la acción Guardar el manifiesto en S3 estuviera habilitada en modo sin conexión cuando se utilizaba. FastLoad
Teradata	RSQL de Amazon Redshift	Se solucionó un problema para garantizar que las reglas de mapeo se aplicaran a scripts como FastLoad.
Greenplum	Amazon Redshift	Se aumentó la versión mínima del controlador admitida para Greenplum a 42.2.5.

Origen	Destino	Novedades, mejoras o correcciones
Greenplum	Amazon Redshift	Se agregó una conexión a Greenplum a través de SSL con la versión 42.2.5 o posterior del controlador.
Oracle DW	Amazon Redshift	Se mejoró la compatibilidad para ejecutar funciones escalares personalizadas definidas por el usuario (UDF) dentro de otra UDF.
Oracle DW	Amazon Redshift	Se solucionó un problema que impedía la aplicación de funciones a la base de datos con este error: `Failed to compile udf` .
Oracle DW	Amazon Redshift	Se mejoró la conversión mediante el uso de las declaraciones de tipo adecuadas, por ejemplo, `pls-type` para parámetros %ROWTYPE.
Teradata	RSQL de Amazon Redshift	Se resolvió un problema que impedía la visualización de los problemas de evaluación del tipo de información en el informe.
Teradata	RSQL de Amazon Redshift	Se resolvió un error del transformador tras convertir algunos scripts.
Teradata	RSQL de Amazon Redshift	Se corrigió un problema que ahora permite agregar comentarios en el script convertido.
Teradata	Amazon Redshift	Se ha resuelto un problema por el que se mostraba FastExport ->EXPORT -> 'null' en lugar de 'CAST' tras la conversión.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba el fallo de algunas funciones de un paquete de extensión al aplicarlas con `Cause:[JDBC Driver]String index out of range: 0` si se utilizaba la versión 1.2.43 del controlador
Teradata	Amazon Redshift	Se añadió la conversión de tablas SET—simulación de tablas SET para instrucciones de inserción y selección.
Teradata	Amazon Redshift	CAST: compatibilidad para tipos de datos adicionales.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se corrigió un error en la conversión de "other_current_time_01"
Teradata	Amazon Redshift	Teradata FastExport — Amazon Redshift RSQL: conversión mejorada de los comandos de Teradata: campo FastExport
Teradata	Amazon Redshift	Teradata FastExport — Amazon Redshift RSQL: conversión mejorada de los comandos de Teradata: diseño FastExport
Oracle	PostgreSQL Aurora PostgreSQL	Se resolvió un problema que provocaba el cambio del script de destino de objetos con la instrucción SAVE EXCEPTIONS tras la reconversión.
Oracle	PostgreSQL Aurora PostgreSQL	Se resolvió un problema que provocaba la especificación de un campo incorrecto en la cláusula ORDER BY tras la conversión a `proc_cursor_with_calc_columns` .
Oracle	PostgreSQL Aurora PostgreSQL	Resuelto: en una conversión ASSOCIATIVE COLLECTION es necesaria una declaración de variables `aws_oracle_ext$array_id$temporary` adicionales.
Oracle	PostgreSQL Aurora PostgreSQL	Resuelto: se produjo una conversión errónea de una CLAVE PRINCIPAL con el mismo nombre que un ÍNDICE que pertenecía a la misma tabla.

Problemas resueltos:

- Mejoras generales.

Notas AWS SCT de publicación de la compilación 654

Origen	Destino	Novedades, mejoras o correcciones
Oracle	PostgreSQL Aurora PostgreSQL	Se resolvió un problema con las pseudocolumnas de consultas jerárquicas y un error al analizar las columnas PRIOR.
Oracle	PostgreSQL Aurora PostgreSQL	Se resolvió un problema que impedía convertir correctamente un comentario de varias líneas que contenía una barra inclinada y un asterisco (/*).
Oracle	PostgreSQL Aurora PostgreSQL	Se agregó la simulación USER_COL_COMMENTS de la vista del sistema al paquete de extensión.
Oracle	PostgreSQL Aurora PostgreSQL	Se mejoró la conversión de literales entre comillas.
DB2 LUW	PostgreSQL	Se mejoró la conversión de instrucciones LABEL que añaden o sustituyen etiquetas en las descripciones de tablas, vistas, alias o columnas.

Origen	Destino	Novedades, mejoras o correcciones
	Aurora PostgreSQL	
Oracle	Ninguna	Se sustituyó la tabla del sistema SYS.USER$ por la vista DBA_USERS y se mejoraron las consultas.
Oracle DW	Amazon Redshift	Se actualizaron las consultas de metadatos de Oracle DW.
Teradata	RSQL de Amazon Redshift	Se agregó soporte para la conversión de scripts shell, Teradata y Teradata FastLoad Basic Teradata Query (BTEQ) a scripts RSQL de Amazon Redshift.
BTEQ de Teradata	RSQL de Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de "merge_01".
BTEQ de Teradata	RSQL de Amazon Redshift	Se resolvió el problema que provocaba la aparición de End or Identify (EOI) al final de un script en una nueva línea.
Azure Synapse	Amazon Redshift	Se mejoró el mensaje de error que aparece cuando se proporciona una contraseña incorrecta para Azure Synapse.
Teradata	Amazon Redshift	Se mejoró la conversión de la instrucción UPDATE para transferir el nombre de alias correcto según el estándar de Teradata.
Teradata	Amazon Redshift	Se resolvió un error de conversión del cursor que impedía recibir acciones.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la eliminación de filas durante una conversión TD_NORMALIZE_OVERLAP.
Teradata	Amazon Redshift	Ahora es compatible con la comprobación estricta de fechas para la función TO_DATE mejorada.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se mejoró la conversión de la funciones integradas TO_NUMBER(n).
Teradata	Amazon Redshift	Se resolvió un problema que impedía la aparición de la categoría Esquemas en el árbol de metadatos.
Greenplum	Amazon Redshift	Se agregó la selección GP_SEGMENT_ID a la lista al crear una partición virtual para una tabla de Greenplum.
Greenplum	Amazon Redshift	Se resolvió un problema que impedía la aplicación de funciones en el destino.
MS SQL Server DW	Amazon Redshift	Se resolvió un problema que provocaba un error de transformación después de la conversión sin el elemento de acción 9996.
MS SQL Server DW	Amazon Redshift	Se resolvió un problema que provocaba el registro de un error al abrir el asistente del paquete de extensión.
MS SQL Server DW	Amazon Redshift	Se resolvió un problema que provocaba el uso de un estilo de comentarios incorrecto en las funciones Redshift Python.
Netezza	Amazon Redshift	Se resolvió un problema por el que no se podía crear un paquete de extensión de Netezza-Redshift con un AWS perfil.
Teradata	RSQL de Amazon Redshift	Se ha mejorado la conversión del comando SESSIONS. FastLoad
Teradata	RSQL de Amazon Redshift	Informes de evaluación de FastLoad scripts mejorados.
Teradata	RSQL de Amazon Redshift	Se implementó la acción Guardar en S3 de FastLoad WRITER.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	RSQL de Amazon Redshift	Se resolvió un problema por el que los botones FastLoad Guardar script y Guardar manifiesto en s3 no estaban activos.
Teradata	RSQL de Amazon Redshift	Se ha resuelto un problema por el que FastLoad multifile_script solo creaba un archivo de manifiesto tras la conversión, en lugar de los tres archivos esperados.
Teradata	RSQL de Amazon Redshift	Se resolvió un problema por el que se FastLoad mostraban carpetas adicionales en una ruta S3.
Teradata	RSQL de Amazon Redshift	Se ha resuelto un problema FastLoad que provocaba que el nombre del archivo de manifiesto fuera incorrecto en una ruta de S3.

Problemas resueltos:

- Mejoras generales.

Notas de la versión de la AWS SCT compilación 653

Origen	Destino	Novedades, mejoras o correcciones
Oracle	PostgreSQL Aurora PostgreSQL	Se implementó la capacidad de convertir el SQL dinámico creado en funciones o procedimientos de llamada.
Oracle	PostgreSQL	Se mejoró la conversión del SQL dinámico: parámetros integrados como variables de enlace.

Origen	Destino	Novedades, mejoras o correcciones
	Aurora PostgreSQL	
Oracle DW 18, 19	Amazon Redshift	Se mejoró la conversión de Oracle a Redshift: mejora de las integraciones de conversión. LISTAGG agregado; LISTAGG analítico.
Oracle DW 18, 19	Amazon Redshift	Se mejoró la conversión de Oracle a Redshift: nuevas características de consulta.
Vertica	Amazon Redshift	Se mejoró la conversión de Vertica a Redshift: conexión de SSL a JDBC con SSL=true.
MS SQL Server DW	Amazon Redshift	Se mejoró la conversión de MS SQL Server a Redshift: tablas externas.
Teradata	Amazon Redshift	Se mejoró la conversión de Teradata a Redshift: tipos de datos INTERVAL, operaciones aritméticas.
Teradata	Amazon Redshift	Se mejoró la conversión de Teradata a Redshift: compatibilidad para alias de columnas laterales.
Oracle	Ninguna	Las siguientes consultas de Loader ahora utilizan `DBA_USERS` en lugar de `SYS.USER$` : • get-tree-path-list- by-name-path .sql • estimate-table-or-view- .sql constraints-by-schema • estimate-table-or-view- .sql constraints-by-selected-schemas
Teradata	Amazon Redshift	Se mejoró la alineación de los comentarios cuando SCT convierte las macros de Teradata en procedimientos almacenados de Redshift.
Oracle DW	Amazon Redshift	Se mejoró la conversión de los elementos del formato de fecha/marca de tiempo: `TO_DATE`, `TO_TIMESTAMP` y `TO_TIMESTAMP_TZ`

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se resolvió el error de conversión del cursor de Teradata.
Teradata	Amazon Redshift	Se resolvió el problema que provocaba la eliminación de atributos de `TD_NORMALIZE_OVERLAP` durante la conversión.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba que se ignorara la función `MAX` cuando SCT convertía una consulta.
Teradata	Amazon Redshift	SCT convierte ahora la función CHARACTERS de Teradata en la función LENGTH de Redshift.
Teradata	Amazon Redshift	SCT ahora admite la conversión de FORMAT a TO_CHAR para los formatos más utilizados.
Todos	Todos	Se mejoró la conversión de rutinas cifradas.

Problemas resueltos:

- Mejoras generales.

Notas de publicación de la AWS SCT compilación 652

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	PostgreSQL	Se añadió el bloqueo de aplicaciones para las funciones `sp_getapplock` y `sp_releaseapplock` .
Ninguna	Amazon Redshift	Se mejoró interfaz de la línea de comandos (CLI): se implementó el modo de comandos de script.
Oracle	PostgreSQL	Se implementó el muestreo de parámetros de rutina dentro de un SQL dinámico.

Origen	Destino	Novedades, mejoras o correcciones
	Aurora PostgreSQL	
Oracle	PostgreSQL Aurora PostgreSQL	Se mejoró la conversión de SQL dinámico creado en funciones o procedimientos de llamada.
Microsoft SQL Server Oracle DB2 LUW	Aurora PostgreSQL	Cada función de lambda se implementa y configura solo una vez a través de una política, y las funciones de lambda comunes se reutilizan en todos los orígenes posibles.
DB2 LUW	PostgreSQL	Se resolvió el problema que provocaba el mensaje de error "9996: Gravedad crítica: se produjo un error en el transformador" al utilizar DB2 LUW como origen.
Teradata	Amazon Redshift	Compatibilidad para expresiones de tablas recursivas en el próximo lanzamiento de Amazon Redshift.
Azure Synapse	Amazon Redshift	Se implementaron reglas de optimización de esquemas.
Teradata	Amazon Redshift	Se implementó compatibilidad para la conversión de zonas horarias de macros de Teradata a procedimientos almacenados de Redshift.
Teradata	Amazon Redshift	Se implementó compatibilidad para conversión aritmética en valores PERIOD.
Teradata	Amazon Redshift	Se implementó compatibilidad para la conversión de expresiones de tablas comunes recursivas de Teradata (RECURSIVE CTE).

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se implementó compatibilidad para identificadores que distingue n entre mayúsculas y minúsculas a través de la configuración del usuario, `enable_case_sensitive_identifier` . Por lo tanto, «COLUMN_NAME» y «Column_Name» se convierten en nombres de columna diferentes.
Teradata	Amazon Redshift	Se resolvió el problema de los tipos de datos decimales, por lo que los campos decimales se convierten ahora con la misma precisión.
Teradata	Amazon Redshift	Se resolvió el problema con la conversión aritmética de intervalos para que la conversión correcta de la resta aritmética de intervalos.
Teradata	Amazon Redshift	Se mejoró la conversión del tipo NUMBER to DATE de Teradata.
Teradata	Amazon Redshift	Se mejoró la conversión del tipo DATE to NUMBER de Teradata
BTEQ de Teradata	Amazon Redshift	Se mejoró la conversión del tipo de datos PERIOD.
Teradata	Amazon Redshift	Se resolvió el problema relacionado con la carga de metadatos de una tabla con columnas GEOMETRY, de manera que ahora se carga correctamente desde Teradata.
Teradata	Amazon Redshift	Se implementó compatibilidad para la conversión de instrucci ones MERGE al convertir macros de Teradata a procedimientos almacenados de Redshift.
Teradata	Amazon Redshift	Se mejoró la conversión de macros simples al migrar de Teradata a Redshift.
Teradata	Amazon Redshift	Se mejoró la conversión de la instrucción UPDATE de Teradata para transferir el nombre de alias correcto según el estándar de Teradata.

Problemas resueltos:

- Mejoras generales.

Notas de publicación de la AWS SCT compilación 651

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	AWS SCT Informes mejorados para actualizar los enlaces a las acciones de conversión recomendadas que se muestran en la lista.
MS SQL Server	PostgreSQL	Se agregó compatibilidad para la conversión de funciones `STR()`.
MS SQL Server	PostgreSQL	Se agregó compatibilidad para convertir el operador EXOR bit a bit (^ en Microsoft SQL Server) a PostgreSQL como operador #.
Oracle	PostgreSQL	Se ha resuelto un problema por el que la `aws_oracle_ext.UNISTR(null)` función del paquete de AWS SCT extensiones se bloqueaba `NULL` en un destino de PostgreSQL. AWS SCT ahora maneja el. `NULL`
BTEQ de Teradata	RSQL de Amazon Redshift	Se mejoró la conversión para resolver un problema en el que la conversión de RSQL MERGE de Amazon Redshift producía un error de transformación.
Oracle DW	Amazon Redshift	Se implementaron integradas mejoradas.
Oracle DW	Amazon Redshift	Se agregaron mejoras basadas en las características de metadatos , que incluyen el particionamiento automático de listas (TBL_PART_LIST_AUTO), la lista de columnas múltiples (TBL_PART_MULTI_LIST) y la referencia de intervalos (TBL_PART_RANGE_INTVAL_REF) .
Ninguno	Amazon Redshift	Se aumentaron los límites de las tablas de particiones físicas utilizadas para las conversiones `UNION ALL`.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se mejoró la conversión en el ámbito de los informes de evaluación.
Teradata	Amazon Redshift	Se mejoró de conversión en conversiones MACRO complejas de Teradata.
Teradata	Amazon Redshift	Se mejoró la conversión de macros de Teradata a procedimientos almacenados de Amazon Redshift y se añadieron comentarios a los SQL no compatibles.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba que la conversión de macros de Teradata a procedimientos almacenados de Amazon Redshift diera lugar a referencias a nombres de alias incorrectas.
Teradata	Amazon Redshift	Se mejoró la conversión de la instrucción `QUALIFY` de Teradata.
Teradata	Amazon Redshift	Se mejoró la conversión para transferir los comentarios a Amazon Redshift y conservar un historial de los cambios realizados en la vista.
Teradata	Amazon Redshift	Se resolvió un problema que provocaba la conversión incorrecta de la cláusula RESET WHEN.
BTEQ de Teradata	Amazon Redshift	Se mejoró la conversión de los scripts de BTEQ que contienen instrucciones MERGE.
Teradata	Amazon Redshift	Se agregaron funciones integradas para mejorar la conversión de los campos de tipos de datos PERIOD.
Microsoft SQL Server	Amazon Redshift	Se mejoró la asignación de tipos de datos de transformación mejorado para el tipo de datos TIME.
Todos	Todos	Se agregó acceso a la publicación inicial del manual Referencia de la CLI de AWS Schema Conversion Tool en formato PDF. Consulte Referencia de la CLI de AWS Schema Conversion Tool.

Problemas resueltos:

- Mejoras generales.

Notas de publicación de la AWS SCT compilación 650

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Se actualizó y se mejoró el uso de los agentes de extracción, lo que incluye: • Una configuración para usar con almacenamiento compartido y un agente de copia dedicado. • Exportación e importación de tareas de extracción de datos de un proyecto a otro. • Compatibilidad para Azure SQL Data Warehouse (Azure Synapse) como origen. • Uso de particionamiento de Netezza nativo. Para obtener más información, consulte Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift.
Todos	Amazon RDS PostgreSQL 13	AWS SCT ahora es compatible con Amazon RDS PostgreSQL 13 como destino.
Microsoft SQL Server	Aurora PostgreSQL	Se mejoró la conversión de un conjunto de resultados de un procedimiento de Microsoft SQL Server a un destino de Aurora PostgreSQL.
Oracle DW	Amazon Redshift	Se mejoró la conversión de Oracle a Amazon Redshift.

Origen	Destino	Novedades, mejoras o correcciones
Oracle DW	Amazon Redshift	Se mejoró la conversión de instrucciones SQL dinámicas.
Oracle DW	Amazon Redshift	Se mejoró la conversión a UDF de SQL.
Oracle DW	Amazon Redshift	Se ha aclarado el mensaje de que AWS SCT no se admite la conversión de tablas externas.
Oracle DW	Amazon Redshift	Se mejoraron las funciones de conversión integradas.
BTEQ de Teradata	RSQL de Amazon Redshift	Se ha mejorado el manejo de los parámetros de sustitución en los scripts de BTEQ al utilizar AWS SCT la GUI.
Microsoft SQL Server DW Microsoft SQL Server Azure Azure Synapse	Todos	Se actualizó la versión mínima del controlador JDBC compatible para Microsoft SQL Server, Azure y Azure Synapse.

Problemas resueltos:

- Teradata: mejoras adicionales de la conversión de macros [RESUELTO]
- Los caracteres se ocultaban en el objetivo y provocaban errores de SQL, por lo que fue necesario rediseñarlos para volver a colocarlos [RESUELTO]
- Mejoras generales

Notas de publicación de la compilación AWS SCT 649

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server DW	Amazon Redshift	Se mejoró la conversión de MSSQL a Amazon Redshift para admitir tablas temporales.
Oracle DW	Amazon Redshift	Se mejoraron las funciones integradas, tales como: Funciones de conversión • TO_BINARY_DOUBLE • TO_BINARY_FLOAT • TO_NUMBER • TO_DATE • TO_TIMESTAMP • TO_TIMESTAMP_TZ • TO_DSINTERVAL • TO_YMINTERVAL • VALIDATE_CONVERSION
Oracle DW	Amazon Redshift	Se mejoraron las funciones para el procesamiento de consultas por aproximación, como: Funciones de agregación • ANY_VALUE • APPROX_COUNT_DISTINCT

Origen	Destino	Novedades, mejoras o correcciones
		• APPROX_COUNT_DISTINCT_DETAIL • APPROX_COUNT_DISTINCT_AGG • LISTAGG • TO_APPROX_COUNT_DISTINCT
Teradata	Amazon Redshift	Se mejoró la conversión para la selección automática de claves de clasificación y distribución de Teradata. El motor de base de datos selecciona automáticamente claves de clasificación y distribución. Se introdujo un botón denominado Utilizar el ajuste automático de tablas de Amazon Redshift al cuadro de diálogo Configuración actual del proyecto > Estrategias de optimización > Estrategia inicial de selección de claves.
Teradata	Amazon Redshift	Cargador de AWS SCT tablas mejorado para garantizar que se AWS SCT carguen todas las tablas de Teradata.
Teradata	Amazon Redshift	Se mejoró la conversión para que Amazon Redshift admita patrones de subconsultas correlacionados que incluyen una cláusula WHERE NOT EXISTS simple.
Teradata	Amazon Redshift	Se agregó compatibilidad para el uso de los comandos ECHO en las macros.
DB2 LUW	PostgreSQL Aurora PostgreSQL	Se implementó la compatibilidad para la conversión de DYNAMIC RESULTS SETS, que incluye: • Cláusula de cursor WITH RETURN/WITH RETURN TO CLIENT • Conversión de cláusula de rutina DYNAMIC RESULT SETS

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server Oracle DB2 LUW SAP ASE	Aurora PostgreSQL	Se implementó la compatibilidad para Aurora RDS PostgreSQL actual como destino.
Microsoft SQL Server Oracle DB2 LUW SAP ASE	MariaDB	Se implementó la compatibilidad para MariaDB 10.5 como destino.
Microsoft SQL Server	MariaDB	Se implementó la compatibilidad para la cláusula INSERT-RETURNING, que devuelve un conjunto de resultados de las filas insertadas.
Oracle	Aurora PostgreSQL	Se agregó compatibilidad para la función XMLFOREST para la conversión de Oracle a Aurora PostgreSQL.

Problemas resueltos:

- Mejoras generales.

Notas de publicación de la compilación AWS SCT 648

Origen	Destino	Novedades, mejoras o correcciones
Oracle	PostgreSQL Edición de Amazon Aurora compatible con PostgreSQL	Se implementó el modo de aplicación personalizado del paquete de extensión de Aurora PostgreSQL: operadores para tipos numéricos/de fecha y texto.
Oracle Microsoft SQL Server DB2 LUW	Aurora PostgreSQL	Se implementó la configuración de invocación de lambda para Aurora PostgreSQL implementada: creación de la extensión aws_lambda; asignación de roles de IAM al clúster de Aurora PostgreSQL. • Oracle: correos electrónicos, trabajos, colas, archivos WebAgent • DB2: correos electrónicos, tareas, archivos • Microsoft SQL Server: correos electrónicos, agente
Oracle	PostgreSQL	Se implementó la refactorización de conversión de instrucciones FORALL: • Instrucción FORALL • FORALL ... SAVE EXCEPTIONS • RETURNING INTO with BULK COLLECT • Colección del sistema SQL%BULK_EXCEPTIONS

Origen	Destino	Novedades, mejoras o correcciones
Oracle DW 18, 19	Amazon Redshift	Se mejoró la conversión de Oracle a Amazon Redshift: mejora de las integraciones de conversión. LISTAGG agregado; LISTAGG analítico .
Oracle DW 18, 19	Amazon Redshift	Se mejoró la conversión de Oracle a Amazon Redshift: nuevas características de consulta.
Vertica	Amazon Redshift	Se mejoró la conversión de Vertica a Amazon Redshift: conexión de SSL a JDBC con SSL=true.
Microsoft SQL Server DW	Amazon Redshift	Se mejoró la conversión de Microsoft SQL Server a Redshift: tablas externas.
Teradata	Amazon Redshift	Se mejoró la conversión de Teradata a Redshift: tipos de datos INTERVAL, operaciones aritméticas.
Teradata	Amazon Redshift	Se mejoró la conversión de Teradata a Redshift: compatibilidad para alias de columnas laterales.

Problemas resueltos:

- Mejoras generales

Notas de publicación de la versión 647 AWS SCT

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SQL Server	Microsoft SQL Server	RDS ahora es compatible con la característica de correo electrónico de base de datos.
Microsoft SQL Server	MySQL	Se implementó el nombre máximo de cada tipo de identificador: la longitud máxima de los nombres de objeto (por ejemplo, tablas, restricciones o columnas) en SQL Server es de 128 caracteres.

Origen	Destino	Novedades, mejoras o correcciones

		La longitud máxima de los nombres de objeto en MySQL es de 64 caracteres. Para escribir los objetos convertidos en la base de datos MySQL, debe acortar sus nombres. Para evitar que los nombres se dupliquen después de cortarlos, es necesario agregar una «suma de comprobación» del nombre del objeto original a los nuevos nombres.
		
		Corte los nombres de más de 64 caracteres de la siguiente manera:
		
		`[first N chars]() + "" + [checksum]()`
		
		`[first N chars] = 64 - 1 - [length of checksum string]`
		
		Por ejemplo: `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`

Oracle	MySQL/ Aurora MySQL	Se implementó la carga y conversión de comentarios en los objetos de almacenamiento. Por ejemplo, el procesamiento de los comentarios en las tablas y el procesamiento de los comentarios en las columnas de tablas o vistas.
Teradata	Amazon Redshift	Se agregó compatibilidad para la conversión de tipos de datos TIME.
Teradata	Amazon Redshift	Se mejoró la conversión: se implementó TD_NORMALIZE_OVERLAP.
Microsoft SQL Server DW	Amazon Redshift	Se mejoró la conversión: SELECT con la cláusula WITH; SELECT sin FROM

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	AWS SCT Asesor del servicio de migración de datos (DMSA): esta nueva función le permite evaluar varios servidores y recibir un informe resumido que muestra la mejor dirección de destino para su entorno.
Todos	Todos	AWS SCT Asistente: la comparación de objetivos ahora muestra las diferencias entre los objetivos en una sola vista de tabla.
Todos	Todos	Interfaz de usuario del filtro de árbol: el filtro de metadatos rediseñad o gestiona patrones de filtrado más complejos.
Todos	Todos	Informe de evaluación: la sección Advertencias rediseñada proporcio na una descripción mejor y una comprensión más clara de los problemas.

Problemas resueltos:

- Mejoras generales
- Extractores de datos: la subtarea falló con ConcurrentModificationException [SOLUCIONADO].
- Microsoft SQL Server a MySQL: longitudes máximas de los identificadores [RESUELTO].

Notas de publicación de la compilación AWS SCT 646

Origen	Destino	Novedades, mejoras o correcciones
Oracle	PostgreSQ L	Se mejoró la implementación del modelo de formato TM.
Oracle	PostgreSQ L	La implementación de la máscara en formato SP proporciona compatibilidad básica con el sufijo SP, solo para el idioma inglés.
Oracle	PostgreSQ L	Manejo de nombres de objetos largos de Oracle: AWS SCT ahora maneja los nombres de objetos largos de Oracle según el atributo de longitud máxima del identificador objetivo.

Origen	Destino	Novedades, mejoras o correcciones
	Amazon Redshift	Codificación AZ64 de Amazon Redshift con AWS SCT : se agregó la codificación de compresión AZ64 para algunos tipos de datos
Teradata	Amazon Redshift	Se agregó compatibilidad para la conversión de transacciones implícitas.
Teradata	Amazon Redshift	Se añadió compatibilidad para las funciones integradas geoespaciales de Teradata: Métodos `ST_LineString`
Greenplum	Amazon Redshift	Conversión secuencial de Greenplum: se añadieron los siguientes elementos a las pestañas de propiedades: valor mínimo, valor máximo, incremento y ciclo.
Greenplum	Amazon Redshift	Solucionador: se agregó la resolución del tipo de datos «char».
Greenplum	Amazon Redshift	Longitud de conversión de caracteres: se actualizó la conversión PL/pgSQL para el tipo de caracteres.
Greenplum	Amazon Redshift	Se solucionó un problema al seleccionar la clave de distribución de Greenplum, que provocaba que una tabla tuviera la clave de distribución pero no AWS SCT pudiera reconocerla ni buscarla como distribuida aleatoriamente.
Teradata	Amazon Redshift	Se añadió compatibilidad para cursores de Teradata: se añadió compatibilidad para la conversión de cursores.
Teradata	Amazon Redshift	Columnas de identidad: se agregó compatibilidad para la conversión de columnas de identidad.
Teradata	Amazon Redshift	Tipos de datos INTERVAL: se agregó compatibilidad para la conversión de tipos de datos INTERVAL.

Problemas resueltos:

- Mejoras generales
- Greenplum: No se pudo ejecutar la conversión debido a un error en el registro [RESUELTO].

- MSSQL — PostgreSQL: error del transformador al convertir la función LAG [RESUELTO].
- MSSQL — PostgreSQL: SCOPE_IDENTITY [RESUELTO].
- AWS SCT bloqueados en los proyectos de DW [SOLUCIONADO].
- Se necesita una regla de mapeo para eliminar el espacio adicional en el nombre de la columna en AWS SCT [RESUELTO].

Notas de publicación de la AWS SCT versión 645

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Proporcione una solución para resolver las vistas no totalmente cualificadas de Teradata (nombres de vistas u objetos no completam ente cualificados dentro de la vista).
Teradata	Amazon Redshift	Se agregó compatibilidad para la función ASCII para calcular los nodos.
Teradata	Amazon Redshift	Cuando AWS SCT detecta datos de varios bytes en un Teradata `CHAR` definido como`CHAR(N)`, se convierte en Amazon `VARCHAR(3 *N)` Redshift.
Teradata	Amazon Redshift	Proporcione una conversión `CAST` de Teradata entre fechas y números. • `SELECT Cast('2020-07-17' AS BIGINT)` • `SELECT Cast(20200630 - 19000000 AS DATE)`
Teradata	Amazon Redshift	Se añadió compatibilidad para la conversión de tipos de datos `PERIOD` de Teradata en dos columnas `TIMESTAMP` de Amazon Redshift: • `PERIOD(TIMESTAMP)` •

Origen	Destino	Novedades, mejoras o correcciones
		`PERIOD(TIMESTAMP WITH TIMEZONE)`
Teradata	Amazon Redshift	Se agregó compatibilidad para la conversión de la función `RANK` de Teradata con una cláusula `RESET WHEN`.
Teradata	Amazon Redshift	Se mejoró la compatibilidad para CAST en las conversiones de tipos de datos explícitos y para CAST implícitos en las expresiones.
Teradata	Amazon Redshift	Informe sobre patrones de subconsultas correlacionadas no compatibles. Para obtener más información, consulte Subconsultas correlacionadas en la Guía para desarrolladores de bases de datos de Amazon Redshift.
ninguno	Amazon Redshift	Se mejoró la compatibilidad del límite de tablas para tipos de nodos RA3.
Teradata	Amazon Redshift	Se agregó compatibilidad para la extracción de datos geoespaciales de Teradata. Para obtener más información, consulte Consultar datos geoespaciales en Amazon Redshift en la Guía para desarrolladores de bases de datos de Amazon Redshift.
Microsoft SQL Server	PostgreSQL	Se agregó la opción `convert_procedures_to_function` .

Problemas resueltos:

- Mejoras generales

Notas de la versión de la compilación 644 AWS SCT

Los cambios de la versión AWS SCT 1.0.643 se han fusionado en la AWS SCT versión 1.0.644.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Múltiples mejoras de conversión. • Se mejoraron las conversiones con `QUALIFY` con el alias de la tabla. • Se mejoraron las conversiones con el operador `IN`. • Se mejoró la conversión con el operador `LIKE`. • Se mejoraron las conversiones al resaltar los problemas en el código convertido. • Se mejoraron las conversiones con un orden inusual de cláusulas `WHERE` y `QUALIFY` en SQL. • Se corrigieron los errores del transformador que se producían durante la conversión de construcciones `JOIN()` del procedimi ento `UPD_FT_SVC_TRANS_BH_CBH_IND` . • Se mejoró la conversión de macros a procedimientos almacenad os. Se agregaron comandos AWS SCT CLI especiales que pueden analizar los scripts sql/bteq proporcionados y generar un informe sobre la cantidad de estructuras sintácticas que se encuentran en el código fuente. • Recuento de comandos de BTEQ • Número de HANDLERS • Número de casos de CAST • Número de casos de DML/DDL

Origen	Destino	Novedades, mejoras o correcciones
		• Número de DML en las vistas actualizables Se agregó un elemento de acción del informe de evaluación: Amazon Redshift no admite columnas de Teradata con formatos de fecha personalizados.
Oracle	PostgreSQL/Aurora PostgreSQL	Se agregó la funcionalidad de guardar los scripts de instalación de los paquetes de extensión. Se modificó el nivel de gravedad del elemento de acción 5334. Se mejoró el rendimiento del uso de un registro como variable `IMPLEMENTATION` de paquete. Se agregó compatibilidad con la función de agregación `XMLAGG`
IBM Db2	PostgreSQL/Aurora PostgreSQL	Se agregó la carga y conversión de comentarios en los objetos de almacenamiento.
MS SQL DW	Amazon Redshift	Se mejoró la conversión: se resolvió el problema con `PATINDEX`. Mejoras de la interfaz de usuario: • Se implementó la opción Guardar como SQL para el árbol de orígenes. • Se añadió lógica a la generación de scripts para varios archivos.
Vertica	Amazon Redshift	Mejora de la interfaz de usuario: se implementó la opción Guardar como SQL para el árbol de orígenes.

Problemas resueltos:

- Se implementaron mejoras generales en las conversiones entre Teradata y Amazon Redshift

- Corrección de errores y mejoras generales de la interfaz de usuario

Notas de publicación de la compilación 6.42 AWS SCT

Cambios en la AWS Schema Conversion Tool versión 1.0.642.

> **Note**
>
> AWS Schema Conversion Tool (AWS SCT) los cambios en la compilación 1.0.642 se aplican a Windows, Ubuntu y Fedora. No existe una compilación 1.0.642 para macOS.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft SSIS	AWS Glue	Se implementó la conversión de los paquetes ETL de Microsoft SQL Server Integration Services (SSIS) a AWS Glue. Para obtener más información, consulte Conversión de SSIS a AWS Glue con AWS SCT.
Oracle	MariaDB/SQL MODE=CLE/MySQL/Amazon Aurora MySQL	Se implementó la sección de declaraciones PL/SQL en la cláusula WITH.
Oracle	PostgreSQL/Aurora PostgreSQL	Se añadió compatibilidad con `DBMS_SESSION.RESET_PACKAGE` y `DBMS_SESSION.MODIFY_PACKAGE` .
Vertica	Amazon Redshift	Habilite la exportación de scripts de SQL de una base de datos de Vertica a Amazon Redshift.

Problemas resueltos:

- Mejora del informe de evaluación.

- Mejora del informe de evaluación de la interfaz de usuario.
- Añada la posibilidad de modificar la configuración de la JVM desde la interfaz de usuario.
- Mejoras generales.

Notas de la versión de la AWS SCT compilación 641

Cambios en la AWS Schema Conversion Tool versión 1.0.641.

> **Note**
>
> AWS Schema Conversion Tool (AWS SCT) los cambios en la compilación 1.0.641 se aplican a Windows, Ubuntu y Fedora. No existe una compilación 1.0.641 para macOS.

Origen	Destino	Novedades, mejoras o correcciones
Oracle/ MS SQL/ MySQL/ PostgreSQL/Db2 LUW	Todos	Genere los cálculos del informe de tiempo en el archivo.csv.
Teradata	Amazon Redshift	Se añadió compatibilidad para la función CSUM. Se añadió compatibilidad para tipos de datos geoespaciales de Teradata.
Teradata	Todos	Se añadió compatibilidad para la conversión de columnas IDENTITY.
Greenplum	Amazon Redshift	Se añadió compatibilidad para el estilo de distribución AUTO durante la conversión de tablas de Greenplum.
SAP ASE	Todos	Genere los cálculos del informe de tiempo en el archivo.csv.

Errores resueltos:

- Correcciones de varios errores.
- Varias mejoras de rendimiento.

Notas de publicación de la compilación 640 AWS SCT

Los cambios de las AWS SCT versiones 1.0.633, 1.0.634, 1.0.635, 1.0.636, 1.0.637, 1.0.638, 1.0.639 y 1.0.640 se han fusionado en la versión 1.0.640. AWS SCT

> **Note**
>
> AWS SCT los cambios en la compilación 1.0.640 se aplican a Windows, Ubuntu y Fedora. No se aplican a MacOS.
> No puedes instalar la AWS SCT versión 1.0.640 o superior en macOS de Apple. AWS SCT La versión 1.0.632 fue la última versión compatible con la instalación en macOS de Apple.

En las tablas siguientes, puede encontrar listas de las características y correcciones de errores para las versiones de AWS Schema Conversion Tool que se han combinado en la versión 1.0.640. Estas tablas agrupan características y correcciones de errores por el motor de origen.

Temas

- Cambios en la versión 1.0.640 de Oracle
- Cambios en la versión 1.0.640 de Microsoft SQL Server
- Cambios en la versión 1.0.640 de MySQL
- Cambios en la versión 1.0.640 de PostgreSQL
- Cambios en la versión 1.0.640 de Db2 LUW
- Cambios en la versión 1.0.640 de Teradata
- Cambios en la versión 1.0.640 para otros motores

Cambios en la versión 1.0.640 de Oracle

En la tabla siguiente se muestran los cambios en la compilación 1.0.640 en los que Oracle es el motor de origen.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	PostgreSQL Aurora PostgreSQL	Se implementó la conversión de código SQL en aplicaciones de Java y Pro*C.
Oracle	PostgreSQL Aurora PostgreSQL	Se mejoró el rendimiento de las siguientes funciones cuando se utilizan en una cláusula WHERE: • aws_oracle_ext.to_date • aws_oracle_ext.to_char • aws_oracle_ext.to_number • aws_oracle_ext.sysdate • aws_oracle_ext.sys_context
Oracle	RDS MariaDB 10.4	Se añadió compatibilidad para RDS MariaDB 10.4 para todos los proveedores de procesamiento transaccional en línea (OLTP).
Oracle	PostgreSQL/Aurora PostgreSQL	Se ha agregado compatibilidad para DBMS_UTILITY.GET_TIME. Se han agregado las siguientes emulaciones: • DBMS_UTILITY.GET_TIME • DBMS_UTILITY.FORMAT_CALL_STACK • DBMS_UTILITY.CURRENT_INSTANCE
Oracle	MariaDB/MySQL/Aurora MySQL/Microsoft SQL Server	Se ha agregado compatibilidad con cláusula de intercambio para TABLE(DATA,EXTENDED DATA), VIEW(DATA,EXTENDED DATA) y SEQUENCE(DATA)

Origen	Destino	Novedades, mejoras o correcciones
	Mode=Oracle/PostgreSQL/Aurora PostgreSQL/RDS Oracle	
Oracle	PostgreSQL/Aurora PostgreSQL/Oracle RDS	La definición DEFAULT de una columna se puede extender para que DEFAULT se aplique para la inserción explícita de NULL. La cláusula DEFAULT tiene una nueva cláusula ON NULL. Esta nueva cláusula indica a la base de datos que asigne un valor de columna predeterminado especificado cuando una instrucción INSERT intenta asignar un valor que se evalúa como NULL.
Oracle	MariaDB/MariaDB (SQL MODE=ORACLE)	Se ha agregado compatibilidad para "Columnas de identidad" que se incrementan automáticamente en el momento de la inserción.
Todos	Todos	Actualice a Amazon Corretto JDK 11 desde JDK 8. Para obtener más información, incluidos los enlaces de descarga, consulte ¿Qué es Amazon Corretto 11? en la Guía del usuario de Amazon Corretto 11.
Todos	Todos	Se agregó información al informe de evaluación sobre posibles incoherencias en la base de datos del usuario.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	MariaDB 10.2/MariaDB 10.3/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	La cláusula `DEFAULT` tiene una cláusula `ON NULL` nueva, que indica a la base de datos que asigne un valor de columna predeterminado especificado cuando una instrucción INSERT intenta asignar un valor que se evalúa como `NULL`.
Oracle	Oracle RDS/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Se agregó compatibilidad para las columnas `IDENTITY`.
Oracle	MySQL 8.x	Se agregó compatibilidad para la restricción CHECK.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	PostgreSQL/Aurora PostgreSQL	Se implementó la comprobación ANYDATA IS NULL/IS NOT NULL mediante la rutina del paquete de extensión. Se implementó la emulación de la función VALUE utilizada en una consulta en función de la función TABLE de XMLSequence. Se agregó compatibilidad con DBMS_LOB para las siguientes rutinas integradas: • DBMS_LOB.CREATETEMPORARY • DBMS_LOB.FREETEMPORARY • DBMS_LOB.APPEND
Todos	SQL Server	SQL Server 2019: se agregó compatibilidad para el nuevo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY. SQL Server 2017: se agregó compatibilidad para los tipos de tabla Node y Edge de bases de datos de Graph. SQL Server 2016: se agregó compatibilidad para TEMPORAL TABLES.
Todos	Todos	Se implementó la capacidad para invalidar particiones físicas con particiones virtuales. Los extractores de almacenamiento de datos extraen datos de acuerdo con las particiones virtuales creadas.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	Amazon Redshift	Se implementó la conversión de atributos de cursor en bloques anidados. Amazon Redshift no admite colecciones. Las variables relacionadas se convierten como VARCHAR. Todas las operaciones de recopilación distintas de la asignación de una variable a otra se rechazan, incluido el acceso a elementos de iniciación y recopilación. Se implementó el estilo de distribución de Amazon Redshift = AUTO.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	PostgreSQL/Aurora PostgreSQL	Si una palabra no reservada en Oracle está reservada en PostgreSQL, se cumple lo siguiente: • Si la palabra está entre comillas, retiene el uso de mayúsculas y minúsculas y se mantiene entre comillas. • Si la palabra no está entre comillas, se pone en mayúsculas y entre comillas. Se ha implementado la capacidad de usar funciones como entrada para las funciones LTRIM, RTRIM y TRIM. Las expresiones SELECT DISTINCT, ORDER BY deben aparecer en la lista de selección. Para los parámetros del cursor que siguen a un parámetro con un valor DEFAULT, AWS SCT añade la cláusula DEFAULT IS NULL Los parámetros del cursor OUT de origen se convierten en parámetros del cursor IN. Se ha vuelto a implementar la variable de paquete agregando la opción "Implementación lógica de variables de paquete" en "Configuración de conversión". Los ajustes disponibles son: "variables de sesión" y "objetos globales plv8". El valor predeterminado es "variables de sesión". Se implementó la compatibilidad de la instrucción AUTONOMOUS_TRANSACTION con dblink y pg_background.
Oracle	Todos	Se implementó la vista SYS_%_TAB_COMMENTS.

Origen	Destino	Novedades, mejoras o correcciones
Oracle	PostgreSQL	Las entradas variables a los filtros no son compatibles con PostgreSQL. Al convertir de Oracle a PostgreSQL, si se encuentra un filtro de variables, ahora se registra una excepción.
Oracle	Amazon Redshift	Se mejoró la conversión del cursor FOR..LOOP en el código almacenado. Se implementó la invocación de código almacenado de funciones o procedimientos con parámetros predeterminados. Se implementó capacidad de código almacenado para UPDATE con alias sin cláusula WHERE. Se implementaron funciones de código almacenado para realizar casos adicionales con SELECT FROM dual. Se implementaron los parámetros y variables de paquete Table %ROWTYPE de código almacenado. Se implementó código almacenado utilizado de JAVA y procedimientos externos. Se implementó el paquete estándar de Oracle en el código almacenado.

Cambios en la versión 1.0.640 de Microsoft SQL Server

En la tabla siguiente se muestran cambios en la compilación 1.0.640 en los que Microsoft SQL Server es el motor de origen.

Origen	Destino	Novedades, mejoras o correcciones
Microsoft Azure/ Microsoft	PostgreSQL/Aurora PostgreSQL/MySQL/	Se agregó compatibilidad para los índices COLUMN STORE.

Origen	Destino	Novedades, mejoras o correcciones
SQL Server	Aurora MySQL	
Microsoft SQL Server	RDS MariaDB 10.4	Se añadió compatibilidad para RDS MariaDB 10.4 para todos los proveedores de procesamiento transaccional en línea (OLTP).
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Se agregó compatibilidad para el atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY.
Azure/SQL Server	MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Se agregó compatibilidad con los tipos de tabla Node y Edge de bases de datos.
Azure/SQL Server	MariaDB/MySQL/Aurora MySQL/PostgreSQL/Aurora PostgreSQL	Se agregó compatibilidad para TEMPORAL TABLES.

Origen	Destino	Novedades, mejoras o correcciones
Todos	Todos	Actualice a Amazon Corretto JDK 11 desde JDK 8. Para obtener más información, incluidos los enlaces de descarga, consulte ¿Qué es Amazon Corretto 11? en la Guía del usuario de Amazon Corretto 11.
Todos	Todos	Se agregó información al informe de evaluación sobre posibles incoherencias en la base de datos del usuario.
Azure/SQL Server	MySQL/ Aurora MySQL/ Pos tgreSQL/ Aurora PostgreSQ L/MariaDB	Se agregó compatibilidad para el procesamiento de DML para SQL Server Graph Architecture.
SQL Server	Aurora PostgreSQ L	Se agregó la opción de convertir parámetros sin el prefijo `par_`.
Azure/SQL Server	MySQL 8.x	Se agregó compatibilidad para la restricción CHECK.
Todos	SQL Server	SQL Server 2019: se agregó compatibilidad para el nuevo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY. SQL Server 2017: se agregó compatibilidad para los tipos de tabla Node y Edge de bases de datos de Graph. SQL Server 2016: se agregó compatibilidad para TEMPORAL TABLES.
Todos	Todos	Se implementó la capacidad para invalidar particiones físicas con particiones virtuales. Los extractores de almacenamiento de datos extraen datos de acuerdo con las particiones virtuales creadas.

Origen	Destino	Novedades, mejoras o correcciones
SQL Server	AWS Glue (Concha de Python)	Mejoras de conversión, incluido: • Se implementó la conversión de funciones integradas en Python.String. • Se implementaron EXECUTE y EXEC en el código almacenado. • Se implementaron usando tipos de tabla.
Azure/SQL Server	PostgreSQL/Aurora PostgreSQL	Se implementó haciendo opcionales los procedimientos $TMP.
SQL Server	MySQL/ Aurora MySQL	Operaciones aritméticas ampliadas con fechas. Emulación de la construcción "TOP" (expresión) WITH TIES. Después de llamar a los procedimientos con el refcursor generado fuera, el refcursor ahora se cierra. La configuración de un nivel de aislamiento GLOBAL no es compatible con Aurora MySQL. Solo se puede cambiar el ámbito de la sesión. El comportamiento predeterminado de las transacciones es utilizar REPEATABLE READ y lecturas consistentes. Es posible que sea necesario modificar las aplicaciones diseñadas para su uso con READ COMMITTED. De forma alternativa, pueden cambiar explícitamente el valor predeterminado a READ COMMITTED.

Origen	Destino	Novedades, mejoras o correcciones
SQL Server	AWS Glue (Concha de Python)	Las instrucciones de SQL Server producen un conjunto de resultados completo, pero hay momentos en los que los resultados se procesan mejor fila por fila. La apertura de un cursor en un conjunto de resultados permite procesar el conjunto de resultados fila por fila. Puede asignar un cursor a una variable o parámetro con un tipo de datos de cursor. Se implementó incluir una serie de instrucciones de Transact-SQL para el código almacenado, de modo que se pueda ejecutar un grupo de instrucciones de Transact-SQL aunque Python no admita las funciones BEGIN y END de SQL Server. control-of-flow Las sentencias LABEL y GOTO de SQL Server no son compatibles con. AWS Glue Si AWS SCT encuentra una etiqueta en el código, se omite. Si AWS SCT encuentra una instrucción GOTO, se comenta.
SQL Server	Amazon Redshift	Se implementó el procesamiento condicional de instrucciones Transact-SQL para código almacenado mediante la implementación del control IF... ELSE. Se implementó la encapsulación de una serie de instrucciones Transact-SQL para el código almacenado de modo que un grupo de instrucciones Transact-SQL se puede ejecutar como un bloque. Admite bloques BEGIN... END anidados. Se han implementado SET y SELECT en el código almacenado. Se implementó CREATE INDEX en Amazon Redshift (que no admite índices) mediante la creación de una clave de clasificación especific ada por el usuario en las tablas.

Cambios en la versión 1.0.640 de MySQL

En la siguiente tabla se muestran los cambios en la compilación 1.0.640 en los que MySQL es el motor de origen.

Origen	Destino	Novedades, mejoras o correcciones
MySQL	PostgreSQL 12.x	Se agregó compatibilidad para columnas generadas.
Todos	Todos	Actualice a Amazon Corretto JDK 11 desde JDK 8. Para obtener más información, incluidos los enlaces de descarga, consulte ¿Qué es Amazon Corretto 11? en la Guía del usuario de Amazon Corretto 11.
Todos	Todos	Se agregó información al informe de evaluación sobre posibles incoherencias en la base de datos del usuario.
MySQL	PostgreSQL/Aurora PostgreSQL 11.	Se añadió compatibilidad para lo siguiente: • Se incrustaron transacciones dentro de procedimientos almacenados en SQL. • La capacidad CALL a los procedimientos almacenados en SQL. • La capacidad de crear procedimientos almacenados en SQL.
Todos	SQL Server	SQL Server 2019: se agregó compatibilidad para el nuevo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY. SQL Server 2017: se agregó compatibilidad para los tipos de tabla Node y Edge de bases de datos de Graph. SQL Server 2016: se agregó compatibilidad para TEMPORAL TABLES.
Todos	Todos	Se implementó la capacidad para invalidar particiones físicas con particiones virtuales. Los extractores de almacenamiento de datos extraen datos de acuerdo con las particiones virtuales creadas.

Cambios en la versión 1.0.640 de PostgreSQL

En la siguiente tabla se muestran los cambios en la compilación 1.0.640 en los que PostgreSQL es el motor de origen.

Origen	Destino	Novedades, mejoras o correcciones
PostgreSQL	MySQL 8.x	MySQL permite ahora crear partes de claves de índice funcionales que indexen valores de expresión en lugar de valores de columna. Las partes clave funcionales permiten la indexación de valores, como los valores JSON, que de otro modo no se pueden indexar. MySQL ahora admite Now CTE y Recursive CTE.
Todos	Todos	Actualice a Amazon Corretto JDK 11 desde JDK 8. Para obtener más información, incluidos los enlaces de descarga, consulte ¿Qué es Amazon Corretto 11? en la Guía del usuario de Amazon Corretto 11.
Todos	Todos	Se agregó información al informe de evaluación sobre posibles incoherencias en la base de datos del usuario.
PostgreSQL 11.x	PostgreSQL/Aurora PostgreSQL 11.	Se añadió compatibilidad para lo siguiente: • Se incrustaron transacciones dentro de procedimientos almacenados en SQL. • La capacidad CALL a los procedimientos almacenados en SQL. • La capacidad de crear procedimientos almacenados en SQL.
PostgreSQL	MySQL 8.x	Se agregó compatibilidad a MySQL para índices descendentes. DESC en una definición de índice ya no se ignora, pero provoca el almacenamiento de los valores clave en orden descendente. Se agregó compatibilidad a MySQL con el uso de expresiones como valores predeterminados en las especificaciones de tipos de datos, incluidas expresiones como valores predeterminados para los tipos de datos BLOB, TEXT, GEOMETRY y JSON. Ahora se pueden utilizar varias funciones de agregación existentes como funciones de ventana. • `AVG()`

Origen	Destino	Novedades, mejoras o correcciones
		• BIT_AND() • BIT_OR() • BIT_XOR() • COUNT() • JSON_ARRAYAGG() • JSON_OBJECTAGG() • MAX() • MIN() • STDDEV_POP() • STDDEV() • STD() • STDDEV_SAMP() • SUM() • VAR_POP() • VARIANCE() • VAR_SAMP() MySQL admite las funciones de ventana que, para cada fila de una consulta, realizan un cálculo utilizando las filas relacionadas con esa fila. • CUME_DIST() • DENSE_RANK() • FIRST_VALUE() • LAG() • LAST_VALUE() • LEAD() • NTH_VALUE() • NTILE()

Origen	Destino	Novedades, mejoras o correcciones
		• PERCENT_RANK() • RANK() • ROW_NUMBER()
PostgreSQL	MySQL 8.x	Se agregó compatibilidad para la restricción CHECK.
Todos	SQL Server	SQL Server 2019: se agregó compatibilidad para el nuevo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY. SQL Server 2017: se agregó compatibilidad para los tipos de tabla Node y Edge de bases de datos de Graph. SQL Server 2016: se agregó compatibilidad para TEMPORAL TABLES.
Todos	Todos	Se implementó la capacidad para invalidar particiones físicas con particiones virtuales. Los extractores de almacenamiento de datos extraen datos de acuerdo con las particiones virtuales creadas.
PostgreSQL/Aurora PostgreSQL	Todos	Se agregó simulación de sysindexes de consulta del sistema. Si hay una instrucción SELECT en un procedimiento sin especificar INTO, se crea el parámetro INOUT p_refcur de tipo refcursor para un procedimiento en el destino.

Cambios en la versión 1.0.640 de Db2 LUW

En la tabla siguiente se muestran los cambios en la compilación 1.0.640 en los que DB2 LUW es el motor de origen.

Origen	Destino	Novedades, mejoras o correcciones
DB2 LUW	RDS MariaDB 10.4	Se añadió compatibilidad para RDS MariaDB 10.4 para todos los proveedores de procesamiento transaccional en línea (OLTP).
Todos	Todos	Actualice a Amazon Corretto JDK 11 desde JDK 8. Para obtener más información, incluidos los enlaces de descarga, consulte ¿Qué es Amazon Corretto 11? en la Guía del usuario de Amazon Corretto 11.
Todos	Todos	Se agregó información al informe de evaluación sobre posibles incoherencias en la base de datos del usuario.
DB2 LUW	MySQL 8.0.17	Se agregó compatibilidad con restricciones CHECK.
Todos	SQL Server	SQL Server 2019: se agregó compatibilidad para el nuevo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY. SQL Server 2017: se agregó compatibilidad para los tipos de tabla Node y Edge de bases de datos de Graph. SQL Server 2016: se agregó compatibilidad para TEMPORAL TABLES.
Todos	Todos	Se implementó la capacidad para invalidar particiones físicas con particiones virtuales. Los extractores de almacenamiento de datos extraen datos de acuerdo con las particiones virtuales creadas.

Cambios en la versión 1.0.640 de Teradata

En la siguiente tabla se muestran los cambios en la compilación 1.0.640 de Teradata.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Amazon Redshift	Se agregó compatibilidad para las instrucciones MERGE y QUALIFY.

Origen	Destino	Novedades, mejoras o correcciones
		Se eliminó la cláusula LOCKING ROWS FOR ACCESS de las instrucciones de Teradata. Se agregó compatibilidad para la función CAST.
Todos	Todos	Actualice a Amazon Corretto JDK 11 desde JDK 8. Para obtener más información, incluidos los enlaces de descarga, consulte ¿Qué es Amazon Corretto 11? en la Guía del usuario de Amazon Corretto 11.
Teradata	Teradata	Se implementaron mejoras en REGEXP_INSTR() y REGEXP_SUBSTR().
Todos	Todos	Se agregó información al informe de evaluación sobre posibles incoherencias en la base de datos del usuario.
Todos	SQL Server	SQL Server 2019: se agregó compatibilidad para el nuevo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY. SQL Server 2017: se agregó compatibilidad para los tipos de tabla Node y Edge de bases de datos de Graph. SQL Server 2016: se agregó compatibilidad para TEMPORAL TABLES.
Teradata	Todos	Se agregó compatibilidad para REGEXP_INSTR() y REGEXP_SUBSTR().
Todos	Todos	Se implementó la capacidad para invalidar particiones físicas con particiones virtuales. Los extractores de almacenamiento de datos extraen datos de acuerdo con las particiones virtuales creadas.
Teradata	Amazon Redshift	Se implementó la capacidad de guardar SQL del árbol de origen en un solo archivo o varios archivos por etapa utilizando la configuración de Project Settings, Save as SQL y Apply, lista desplegable: un solo archivo o varios archivos. Se mejoraron las conversiones de consultas y procedimientos.

Origen	Destino	Novedades, mejoras o correcciones
Teradata	Todos	Se agregó compatibilidad para la versión 16.20 de Teradata

Cambios en la versión 1.0.640 para otros motores

En la siguiente tabla se muestran los cambios en la compilación 1.0.640 para otros motores de origen.

Origen	Destino	Novedades, mejoras o correcciones
Sybase	RDS MariaDB 10.4	Se añadió compatibilidad para RDS MariaDB 10.4 para todos los proveedores de procesamiento transaccional en línea (OLTP).
SAP ASE	MariaDB	Se implementó lo siguiente: • MariaDB 10.4 • Instrucción EXECUTE IMMEDIATE • Definiciones DEFAULT • Compatibilidad con restricciones CHECK
SAP ASE	PostgreSQL 12.x	Se agregó compatibilidad para columnas generadas.
Todos	Todos	Actualice a Amazon Corretto JDK 11 desde JDK 8. Para obtener más información, incluidos los enlaces de descarga, consulte ¿Qué es Amazon Corretto 11? en la Guía del usuario de Amazon Corretto 11.
Todos	Todos	Se agregó información al informe de evaluación sobre posibles incoherencias en la base de datos del usuario.
SAP ASE	MySQL 8.0.17	Se agregó compatibilidad con restricciones CHECK.
Todos	SQL Server	

Origen	Destino	Novedades, mejoras o correcciones
		SQL Server 2019: se agregó compatibilidad para el nuevo atributo de índice OPTIMIZE_FOR_SEQUENTIAL_KEY. SQL Server 2017: se agregó compatibilidad para los tipos de tabla Node y Edge de bases de datos de Graph. SQL Server 2016: se agregó compatibilidad para TEMPORAL TABLES.
Vertica	Amazon Redshift	Se agregó compatibilidad para el estilo de distribución = AUTO.
Todos	Todos	Se implementó la capacidad para invalidar particiones físicas con particiones virtuales. Los extractores de almacenamiento de datos extraen datos de acuerdo con las particiones virtuales creadas.
Amazon Redshift	Amazon Redshift	Las funciones integradas no admitidas en instrucciones DML se reemplazan por NULL como un marcador de posición.
Sybase	PostgreSQL	Se agregó compatibilidad para funciones nativas.
SAP ASE	MySQL/ Aurora MySQL	El nivel de aislamiento predeterminado para Aurora MySQL es REPEATABLE READ. La configuración de un nivel de aislamien to GLOBAL no es compatible con Aurora MySQL. Solo se puede cambiar el ámbito de la sesión. El comportamiento predeterm inado de las transacciones es utilizar REPEATABLE READ y lecturas consistentes. Es posible que sea necesario modificar las aplicaciones diseñadas para ejecutarse con READ COMMITTED. O puede cambiar explícitamente el valor predeterminado a READ COMMITTED.
SAP ASE	PostgreSQL	Se agregó compatibilidad para la función CONVERT (optimista) sin el paquete de extensión.

Origen	Destino	Novedades, mejoras o correcciones
SAP ASE	Todos	Se agregó simulación de sysindexes de consulta del sistema. Si hay una instrucción SELECT en un procedimiento sin especificar INTO, se crea el parámetro INOUT p_refcur de tipo refcursor para un procedimiento en el destino.
Greenplum	Amazon Redshift	Se implementó CREATE TEMPORARY TABLE de la siguiente manera:

Historial de documentos

En la siguiente tabla se describen los cambios importantes en la guía del usuario de AWS Schema Conversion Tool (AWS SCT) después de enero de 2018.

Puede suscribirse a una fuente RSS para recibir notificaciones de las actualizaciones de esta documentación.

Cambio	Descripción	Fecha
Compilación 1.0.672 de AWS SCT	La compilación 1.0.672 proporciona compatibilidad con Amazon RDS para PostgreSQL 15 como destino y Microsoft SQL Server versión 2022 como origen. También agrega compatibilidad con las nuevas características de Amazon Redshift en el código convertido, implementa varias mejoras de conversión para el código fuente de IBM Db2 para z/OS y resuelve varios problemas de conversión.	8 de mayo de 2023
Compilación 1.0.671 de AWS SCT	La compilación 1.0.671 proporciona compatibilidad con las migraciones de Apache Oozie a AWS Step Functions. También agrega compatibilidad con BigQuery como origen para el proceso de evaluación multiservidor. Además, agrega una nueva configuración de conversión para IBM Db2 para z/OS	8 de marzo de 2023

	como origen y resuelve varios problemas de conversión.	
Compilación 1.0.670 de AWS SCT	La compilación 1.0.670 proporciona compatibilidad con las migraciones de Hadoop a Amazon EMR. También agrega compatibilidad con Azure Synapse Analytics como origen para el proceso de evaluación multiservidor. Asimismo, mejora la conversión del código SQL integrado en las aplicaciones de Java y resuelve una serie de problemas de conversión.	23 de enero de 2023
Compilación 1.0.669 de AWS SCT	La compilación 1.0.669 implementa particiones nativas para la migración de datos desde los almacenamientos de datos de Oracle. También mejora el proceso de evaluación multiservidor, agrega nuevas características a los agentes de extracción de datos y resuelve una serie de problemas de conversión.	19 de diciembre de 2022

Compilación 1.0.668 de AWS SCT	La compilación 1.0.668 implementa particionamiento virtual automático para la migración de datos desde las bases de datos de Greenplum y agrega compatibilidad con la migración de datos desde las bases de datos de Snowflake a Amazon Redshift. Además, mejora la conversión del código SQL integrado en las aplicaciones de C# y resuelve una serie de problemas de conversión.	16 de noviembre de 2022
Compilación 1.0.667 de AWS SCT	La compilación 1.0.667 agrega compatibilidad con el motor de extracción, transformación y carga (ETL) de Informatica como origen de migración. También actualiza la versión del paquete de extensión, aumenta la versión mínima del controlador compatible con Amazon Redshift y resuelve varios problemas de conversió n.	13 de octubre de 2022

Compilación 1.0.666 de AWS SCT	La compilación 1.0.666 mejora la conversión de las aplicaciones de Java al agregar soporte para el marco MyBatis. También agrega nuevas funciones a los paquetes de extensión, mejora el cargador de metadatos de origen y resuelve una serie de problemas de conversión.	20 de septiembre de 2022
Compilación 1.0.665 de AWS SCT	La compilación 1.0.665 admite BigQuery como origen de migración. La compilaci ón 1.0.669 permite admitir particiones nativas para la migración de datos desde los almacenamientos de datos de Oracle. Además, mejora la conversión de los almacenam ientos de datos a Amazon Redshift y resuelve varios problemas de conversión.	29 de agosto de 2022

Compilación 1.0.664 de AWS SCT	La compilación 1.0.664 proporciona compatibilidad para Amazon Redshift sin servidor como origen o destino de migración. También implementa el equilibrado automático de memoria en las tareas de extracción de datos y corrige un error que impedía a AWS SCT conectarse a los dispositivos AWS Snowball. Además, agrega la posibilidad de cambiar la intercalación de columnas en las reglas de migración, mejora la interfaz de usuario y resuelve una serie de problemas de conversión.	14 de julio de 2022
Compilación 1.0.663 de AWS SCT	La compilación 1.0.663 agrega compatibilidad con Babelfish para Aurora PostgreSQL 1.2.0 y mejora las capacidades de los informes de evaluación multiservidor. También agrega nuevas características a las reglas de migración, corrige dos errores de carga y resuelve varios problemas de conversión.	20 de junio de 2022

Compilación 1.0.662 de AWS SCT	La compilación 1.0.662 implementa la conversión de código SQL en aplicaciones de C# y mejora el flujo de trabajo de los informes de evaluación multiservidor. También agrega varias mejoras de conversión y resuelve varios problemas de conversión.	19 de mayo de 2022
Compilación 1.0.661 de AWS SCT	La compilación 1.0.661 proporciona compatibilidad con IBM Db2 para z/OS como origen de migración. También agrega compatibilidad con la conversión de scripts de extracción, transformación y carga (ETL) a AWS Glue Studio y resuelve una serie de problemas de conversión.	21 de abril de 2022
Compilación 1.0.660 de AWS SCT	La compilación 1.0.660 proporciona compatibilidad con la versión principal 14 de PostgreSQL y MariaDB 10.6 como destinos de la migración. También agrega la conversión de los índices de Oracle a vistas materializadas y resuelve varios problemas de conversión.	21 de marzo de 2022

Compilación 1.0.659 de AWS SCT	La compilación 1.0.659 proporciona compatibilidad con la versión principal 13 de Aurora PostgreSQL-Compatible Edition como destino de migración. Implementa la conversión de código SQL en aplicaciones de C#, agrega compatibilidad con Oracle Unified Auditing y resuelve varios problemas de conversión.	21 de febrero de 2022
Compilación 1.0.658 de AWS SCT	La compilación 1.0.658 proporciona integración con AWS Secrets Manager y agrega compatibilidad con la plataforma de base de datos de destino virtual Amazon Redshift. También agrega una serie de mejoras de conversión y correcciones de errores.	20 de enero de 2022
Compilación 1.0.657 de AWS SCT	La compilación 1.0.657 mejora la conversión de Microsoft SQL Server a Aurora PostgreSQL-Compatible Edition, Amazon RDS para PostgreSQL y otros destinos de migración. También agrega una serie de mejoras de la interfaz de usuario y correcciones de errores.	20 de diciembre de 2021

Compilación 1.0.656 de AWS SCT	La compilación 1.0.656 proporciona compatibilidad con múltiples bases de datos de origen y destino en un proyecto. También agrega una serie de mejoras de conversió n, estrategia de optimización, mejoras generales y correccio nes de errores.	22 de noviembre de 2021
Compilación 1.0.655 de AWS SCT	La compilación 1.0.655 implementa la conversión de los scripts de trabajo de FastExport de Teradata a RSQL de Amazon Redshift y aumenta la versión mínima del controlador compatible con Greenplum a 42.2.5. También agrega una serie de mejoras y correcciones de errores.	18 de octubre de 2021
Compilación 1.0.654 de AWS SCT	La compilación 1.0.654 implementa la conversión de scripts de Shell, FastLoad de Teradata y Basic Teradata Query (BTEQ) de Teradata a RSQL de Amazon Redshift. También resuelve una serie de problemas de conversión y agrega una serie de correccio nes de errores.	16 de septiembre de 2021

Compilación 1.0.653 de AWS SCT	La compilación 1.0.653 implementa la conversión de SQL dinámico creado en funciones o procedimientos de llamada. También mejora la conversión de rutinas cifradas y agrega una serie de mejoras y correcciones de errores.	10 de agosto de 2021
Compilación 1.0.652 de AWS SCT	La compilación 1.0.652 implementa el modo de comandos de script en la interfaz de la línea de comandos e implementa reglas de optimización de esquemas. También agrega una serie de mejoras de conversión y rendimiento y correcciones de errores.	30 de junio de 2021
Compilación 1.0.651 de AWS SCT	La compilación 1.0.651 agrega una serie de mejoras y correcciones de errores. También proporciona acceso a la copia inicial de la Referenci a de la CLI de AWS Schema Conversion Tool.	4 de junio de 2021

Compilación 1.0.650 de AWS SCT	La compilación 1.0.650 proporciona compatibilidad con Amazon RDS para PostgreSQL 13 como base de datos de destino y actualiza los agentes de extracción. También actualiza la versión mínima del controlador JDBC compatible para Microsoft SQL Server, Azure y Azure Synapse. Además, agrega una serie de mejoras de conversión y correcciones de errores.	30 de abril de 2021
Compilación 1.0.649 de AWS SCT	La compilación 1.0.649 implementa la compatibilidad con MariaDB 10.5 como base de datos de destino e implementa mejoras de funciones para la conversión de las funciones integradas de Oracle. También agrega una serie de mejoras de conversión y rendimiento y correcciones de errores.	29 de marzo de 2021
Compilación 1.0.648 de AWS SCT	La compilación 1.0.648 agrega una serie de mejoras de conversión y correcciones de errores.	22 de febrero de 2021

Compilación 1.0.647 de AWS SCT	La compilación 1.0.647 agrega compatibilidad con la característica de correo electrónico de base de datos en Amazon RDS e implementa la carga y la conversión de comentarios en los objetos de almacenamiento. También incorpora el asesor del servicio de migración de datos de AWS SCT y el asistente de AWS SCT, e implementa la interfaz de usuario con filtros de árbol. Además, agrega una sección rediseñada en el informe de evaluación y una serie de mejoras y correcciones de errores.	15 de enero de 2021
Compilación 1.0.646 de AWS SCT	La compilación 1.0.646 agrega compatibilidad con los tipos de datos INTERVAL, las columnas de identidad y la conversión de cursores, y agrega una serie de mejoras y correcciones de errores.	28 de diciembre de 2020
Compilación 1.0.645 de AWS SCT	La compilación 1.0.645 agrega compatibilidad con la conversión de ETL SSIS a AWS Glue y agrega una serie de mejoras y correcciones de errores.	16 de noviembre de 2020

Compilaciones 1.0.643-1.0.644 de AWS SCT	La compilación 1.0.644 agrega una serie de mejoras de conversión, rendimiento e interfaz de usuario, y correcciones de errores.	14 de octubre de 2022
Compilación 1.0.642 de AWS SCT	La compilación 1.0.642 implementa la conversión de paquetes ETL de Microsoft SQL Server Integration Services a AWS Glue y agrega una serie de mejoras y correcciones de errores.	28 de agosto de 2020
Compilación 1.0.641 de AWS SCT	Esta compilación agrega compatibilidad SSL para extractores de datos. También agrega una serie de mejoras y correcciones de errores.	17 de julio de 2020
Compilaciones 1.0.633-1.0.640 de AWS SCT	Estas compilaciones actualizan de JDK 8 a Amazon Corretto JDK 11. Se han agregado tablas que identifican otras actualizaciones, cambios y correcciones.	22 de junio de 2020
Disponibilidad de AWS WQF	AWS SCT ya no proporciona la herramienta AWS Workload Qualification Framework (AWS WQF) para su descarga.	19 de junio de 2020

Compilación 1.0.632 de AWS SCT	Interfaz de usuario de SCT: se ha agregado una nueva pestaña para mostrar los errores que se producen al aplicar scripts. Ahora puede guardar el árbol de origen como SQL al convertir desde SAP ASE. Mejoras para las conversiones a PostgreSQL, Aurora PostgreSQL o Redshift.	19 de noviembre de 2019
Compilaciones 1.0.631 y 1.0.630 (combinadas) de AWS SCT	Se ha mejorado la compatibilidad con ROWID en Oracle y con los objetos del sistema en Microsoft SQL Server y SAP ASE. Se ha mejorado el tratamiento de los especificadores que faltan en los esquemas de SQL Server. Se ha mejorado la compatibilidad con las conversiones de Greenplum a Redshift. Se ha mejorado la compatibilidad con la conversión de código almacenado al migrar a Amazon Redshift, MariaDB, MySQL y PostgreSQL.	30 de septiembre de 2019

Compilación 1.0.629 de AWS SCT	Compatibilidad con procedimientos almacenados para conversiones de Netezza. Se ha mejorado la compatibilidad con las conversiones a Amazon Redshift, DynamoDB, MySQL y PostgreSQL. Se ha añadido compatibilidad con SAP ASE 12.5 como origen.	20 de agosto de 2019
Compilación 1.0.628 de AWS SCT	Compatibilidad con la emulación de servicios para conversiones de DB2, SQL Server y Oracle. Mejoras de las conversiones a Amazon Redshift, incluida más compatibilidad con cursores y procedimientos almacenados.	22 de junio de 2019
Compilación 1.0.627 de AWS SCT	Compatibilidad con las conversiones de SQL Server a procedimientos almacenados en Amazon Redshift. Mejoras en las conversiones a PostgreSQL 11 y MySQL 8.0.	31 de mayo de 2019
Compilación 1.0.626 de AWS SCT	PostgreSQL 11 y MySQL 8.0 son ahora destinos compatibles. SAP ASE 15.5 ya se admite como origen.	26 de abril de 2019

Compilación 1.0.625 de AWS SCT	Las actualizaciones incluyen la capacidad de convertir BTEQ de Teradata a AWS Glue, compatibilidad con conversiones a MariaDB 10.3 con modo de compatibilidad con Oracle, compatibilidad con SAP ASE 15.7 y sustituciones de servicio para emular la funcionalidad que falta.	25 de marzo de 2019
Compilación 1.0.624 de AWS SCT	Las actualizaciones incluyen la capacidad de convertir ETL de Oracle a AWS Glue, y compatibilidad de conversiones de Microsoft SQL Server, Oracle e IBM Db2 LUW a Amazon RDS para MariaDB. También se ha agregado compatibilidad con conversiones de SAP ASE a RDS for MySQL y Amazon Aurora con compatibilidad con MySQL. Además, hemos añadido compatibilidad con la extensión Orafce durante la conversión de Oracle a PostgreSQL.	22 de febrero de 2019

Compilación 1.0.623 de AWS SCT	Las actualizaciones incluyen la capacidad de convertir las bases de datos de SAP ASE y la capacidad de convertir scripts de T-SQL, DML y DDL en código o componentes equivalentes. También se han añadido emulaciones de Oracle y Microsoft SQL Server para mejorar las conversiones.	25 de enero de 2019
Compilación 1.0.622 de AWS SCT	Las actualizaciones incluyen Workload Qualification Framework, que analiza la carga de trabajo de la migración completa, incluida la base de datos y las modificaciones de la aplicación.	20 de diciembre de 2018
Compilación 1.0.621 de AWS SCT	Entre las actualizaciones se incluyen la compatibilidad con Aurora PostgreSQL 10 como destino y la capacidad de migrar desde Netezza utilizando opciones de tablas externas.	21 de noviembre de 2018
Compilación 1.0.620 de AWS SCT	Entre las actualizaciones se incluyen la capacidad de guardar scripts SQL y compatibilidad con cursores globales de Oracle al migrar a MySQL.	22 de octubre de 2018

Compilación 1.0.619 de AWS SCT	Las actualizaciones incluyen soporte para migrar desde Apache Cassandra a DynamoDB y compatibilidad con Vertica 9 como origen.	20 de septiembre de 2018
Compilación 1.0.618 de AWS SCT	Las actualizaciones incluyen informes de evaluación ampliados, compatibilidad con convertir las pseudocolumnas ROWID de Oracle y compatibilidad con las tablas definidas por el usuario de SQL Server.	24 de agosto de 2018
Compilación 1.0.617 de AWS SCT	Las actualizaciones incluyen informes de evaluación ampliados, compatibilidad con convertir las pseudocolumnas ROWID de Oracle y compatibilidad con las tablas definidas por el usuario de SQL Server.	24 de julio de 2018
Compilación 1.0.616 de AWS SCT	Las actualizaciones incluyen compatibilidad con RDS al convertir de Oracle a Amazon RDS para Oracle y al convertir objetos de programación de Oracle, así como compatibilidad con trabajos de Oracle. Particiones y Db2 LUW versión 10.1.	26 de junio de 2018

Compilación 1.0.615 de AWS SCT	Las actualizaciones permite convertir SQL Server a instrucciones GOTO de PostgreSQL, particiones de PostgreSQL 10 y Db2 LUW versión 10.1.	24 de mayo de 2018
Compilación 1.0.614 de AWS SCT	Las actualizaciones permite convertir de Oracle a Oracle DB Links, de SQL Server a funciones insertadas de PostgreSQL y emulación de objetos del sistema de Oracle.	25 de abril de 2018
Compilación 1.0.613 de AWS SCT	Las actualizaciones incluyen compatibilidad con Db2 LUW, conversión de archivos de SQL*Plus y autenticación de Windows en SQL Server.	28 de marzo de 2018
Compilación 1.0.612 de AWS SCT	Las actualizaciones incluyen compatibilidad con la asignación de tipos de datos personalizados, comparación de esquemas de Oracle 10 y conversión de variables globales de Oracle a PostgreSQL.	22 de febrero de 2018
Compilación 1.0.611 de AWS SCT	Las actualizaciones incluyen compatibilidad de Oracle con instrucciones dinámicas de PostgreSQL, la apertura de un archivo log seleccion ando un mensaje de error y la capacidad de ocultar esquemas en la vista de árbol.	23 de enero de 2018

Actualizaciones anteriores

En la siguiente tabla se describen los cambios importantes en la guía del usuario de AWS Schema Conversion Tool (AWS SCT) antes de enero de 2018.

Versión	Cambio	Descripción	Fecha de modificación
1.0.608	Compatibilidad con puntos de conexión de FIPS para Amazon S3	Ahora puede solicitar a AWS SCT que se conecte a Amazon S3 y Amazon Redshift usando puntos de conexión de FIPS si tiene que cumplir con los requisitos de seguridad del Estándar federal de procesamiento de la información. Para obtener más información, consulte Almacenamiento de credenciales de AWS.	17 de noviembre de 2017
1.0.607	Compatibilidad con puntos de conexión de FIPS para Amazon S3	Ahora puede solicitar a AWS SCT que se conecte a Amazon S3 y Amazon Redshift usando puntos de conexión de FIPS si tiene que cumplir con los requisitos de seguridad del Estándar federal de procesamiento de la información. Para obtener más información, consulte Almacenamiento de credenciales de AWS.	30 de octubre de 2017
1.0.607	Las tareas de extracción de datos pueden hacer caso omiso de LOB	Al crear tareas de extracción de datos, ahora puede optar por omitir los objetos grandes (LOB) para reducir la cantidad de datos que va a extraer. Para obtener más información, consulte Crear, ejecutar y supervisar una tarea de extracción de AWS SCT datos.	30 de octubre de 2017
1.0.605	Acceso al registro de tareas del agente de	A partir de ahora, puede obtener acceso al registro de tareas del agente de extracció n de datos desde un práctico enlace en la interfaz de usuario de AWS Schema	28 de agosto de 2017

Versión	Cambio	Descripción	Fecha de modificación
	extracción de datos	Conversion Tool. Para obtener más información, consulte Crear, ejecutar y supervisar una tarea de extracción de AWS SCT datos.	
1.0.604	Mejoras del convertidor	El motor de AWS Schema Conversion Tool ha sido mejorado para ofrecer conversiones más eficaces en migraciones heterogéneas.	24 de junio de 2017
1.0.603	Filtros de soporte de agentes de extracción de datos	A partir de ahora, podrá filtrar los datos que extraen los agentes de extracción de su data warehouse. Para obtener más información, consulte Crear reglas de migración de datos en AWS SCT.	16 de junio de 2017
1.0.603	AWS SCT es compatible con versiones adicionales de almacenes de datos	A partir de ahora, puede utilizar AWS Schema Conversion Tool para convertir sus esquemas de Teradata 13 y Oracle Data Warehouse 10 a esquemas de Amazon Redshift equivalentes. Para obtener más información, consulte Convertir esquemas de almacenamiento de datos a Amazon Redshift mediante AWS SCT.	16 de junio de 2017
1.0.602	Los agentes de extracción de datos son compatibles con almacenamientos de datos adicionales	A partir de ahora puede usar agentes de extracción de datos para extraer datos desde sus almacenamientos de datos de Microsoft SQL Server. Para obtener más información, consulte Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift.	11 de mayo de 2017

Versión	Cambio	Descripción	Fecha de modificación
1.0.602	Los agentes de extracción de datos pueden copiar datos en Amazon Redshift	Ahora, los agentes de extracción de datos tienen tres modelos de carga. A partir de ahora, puede especificar si desea simplemente extraer los datos, extraerlos y cargarlos en Amazon S3 o extraerlos, cargarlos y copiarlos directamente en Amazon Redshift. Para obtener más información, consulte Crear, ejecutar y supervisar una tarea de extracción de AWS SCT datos.	11 de mayo de 2017
1.0.601	AWS SCT es compatible con almacenamientos de datos adicionales	A partir de ahora, puede utilizar AWS Schema Conversion Tool para convertir sus esquemas de Vertica y Microsoft SQL Server a esquemas de Amazon Redshift equivalentes. Para obtener más información, consulte Convertir esquemas de almacenamiento de datos a Amazon Redshift mediante AWS SCT.	18 de abril de 2017
1.0.601	Los agentes de extracción de datos son compatibles con almacenamientos de datos adicionales	A partir de ahora puede usar agentes de extracción de datos para extraer datos desde sus almacenamientos de datos de Greenplum, Netezza y Vertica. Para obtener más información, consulte Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift.	18 de abril de 2017

Versión	Cambio	Descripción	Fecha de modificación
1.0.601	Los agentes de extracción de datos son compatibles con sistemas operativos adicionales	A partir de ahora, puede instalar agentes de extracción de datos en equipos que ejecuten los sistemas operativos macOS y Microsoft Windows. Para obtener más información, consulte Instalación de agentes de extracción.	18 de abril de 2017
1.0.601	Los agentes de extracción de datos realizan cargas a Amazon S3 automáticamente	A partir de ahora, los agentes de extracción de datos cargan sus datos extraídos en Amazon S3 automáticamente. Para obtener más información, consulte Resultado de la tarea de extracción de datos.	18 de abril de 2017
1.0.600	Agentes de extracción de datos	A partir de ahora, puede instalar agentes de extracción de datos que extraen los datos de su almacenamiento de datos y prepararlos para su uso con Amazon Redshift. Puede utilizar AWS Schema Conversion Tool para registrar los agentes y crear las tareas de extracción de datos para los mismos. Para obtener más información, consulte Migración de datos desde un almacenamiento de datos en las instalaciones a Amazon Redshift.	16 de febrero de 2017

Versión	Cambio	Descripción	Fecha de modificación
1.0.600	Opinión de cliente	Ahora puede expresar sus comentarios sobre la AWS Schema Conversion Tool. Puede archivar un informe de errores, enviar una solicitud de función o proporcionar información general. Para obtener más información, consulte Enviar comentarios.	16 de febrero de 2017
1.0.502	Integración con AWS DMS	A partir de ahora, puede utilizar AWS Schema Conversion Tool para crear puntos de conexión y tareas AWS DMS. Puede ejecutar y monitorizar las tareas desde la AWS SCT. Para obtener más información, consulte Uso de AWS SCT con AWS DMS.	20 de diciembre de 2016
1.0.502	Amazon Aurora compatible con PostgreSQL como base de datos de destino	A partir de ahora, la AWS Schema Conversion Tool admite Amazon Aurora compatible con PostgreSQL como base de datos de destino. Para obtener más información, consulte Conversión de esquemas de base de datos mediante AWS SCT.	20 de diciembre de 2016
1.0.502	Compatibilidad con perfiles	A partir de ahora, puede almacenar diferentes perfiles en la AWS Schema Conversion Tool y cambiar fácilmente entre ellos. Para obtener más información, consulte Almacenamiento de perfiles de servicio de AWS en AWS SCT.	20 de diciembre de 2016

Versión	Cambio	Descripción	Fecha de modificación
1.0.501	Compatibilidad con Greenplum Database y Netezza	A partir de ahora, puede utilizar AWS Schema Conversion Tool para convertir sus esquemas de almacenamiento de datos de Greenplum Database y Netezza a Amazon Redshift. Para obtener más información, consulte Convertir esquemas de almacenamiento de datos a Amazon Redshift mediante AWS SCT.	17 de noviembre de 2016
1.0.501	Optimización de Redshift	Ahora puede utilizar AWS Schema Conversion Tool para optimizar sus bases de datos de Amazon Redshift. Para obtener más información, consulte Optimizar Amazon Redshift mediante AWS SCT.	17 de noviembre de 2016
1.0.500	Reglas de asignación	A partir de ahora, antes de convertir su esquema con la AWS Schema Conversion Tool, puede configurar reglas que cambien el tipo de datos de las columnas, mover objetos de un esquema a otro y cambiar los nombres de los objetos. Para obtener más información, consulte Crear reglas de migración en AWS SCT.	4 de octubre de 2016

Versión	Cambio	Descripción	Fecha de modificación
1.0.500	Migración a la nube	Ahora también puede utilizar AWS Schema Conversion Tool para copiar un esquema de base de datos en las instalaciones existente a una instancia de base de datos de Amazon RDS que ejecute el mismo motor. Puede utilizar esta característica para analizar posibles ahorros de costos en la migración a la nube y en el cambio del tipo de licencia. Para obtener más información, consulte Crear informes de evaluación de migración con AWS SCT.	4 de octubre de 2016
1.0.400	Conversiones de esquemas de almacenamientos de datos	A partir de ahora, puede utilizar AWS Schema Conversion Tool para convertir sus esquemas de almacenamiento de datos de Oracle y Teradata a Amazon Redshift. Para obtener más información, consulte Convertir esquemas de almacenamiento de datos a Amazon Redshift mediante AWS SCT.	13 de julio de 2016
1.0.400	Conversiones de SQL en aplicaciones	A partir de ahora, puede utilizar AWS Schema Conversion Tool para convertir SQL en su código C++, C#, Java u otro tipo de código de aplicación. Para obtener más información, consulte Conversión de SQL de las aplicaciones mediante AWS SCT.	13 de julio de 2016

Versión	Cambio	Descripción	Fecha de modificación
1.0.400	Nueva característica	AWS Schema Conversion Tool ahora contiene un paquete de extensión y un asistente para ayudarle a instalar, crear y configurar las funciones de AWS Lambda y las bibliotecas de Python para proporcionar funcionalidades de correo electrónico, programación de trabajos y otras. Para obtener más información, consulte Uso de las AWS Lambda funciones del paquete de AWS SCT extensión y Uso de bibliotecas personalizadas para los paquetes de extensión de AWS SCT.	13 de julio de 2016
1.0.301	SSL Support	Ahora puede usar la capa de conexión segura (SSL) para conectarse a su base de datos de origen cuando utilice la AWS Schema Conversion Tool.	19 de mayo de 2016
1.0.203	Nueva característica	Agrega soporte para MySQL y PostgreSQL como bases de datos de origen para las conversiones.	11 de abril de 2016
1.0.202	Versión de mantenimiento	Agrega soporte para editar el SQL convertido o que se haya generado para el motor de base de datos de destino. Agrega capacidades de selección mejoradas a las vistas en árbol de la base de datos de origen y de la instancia de base de datos de destino. Agrega soporte para conectarse a una base de datos Oracle de origen con nombres del Transparent Network Substrate (TNS).	2 de marzo de 2016

Versión	Cambio	Descripción	Fecha de modificación
1.0.200	Versión de mantenimiento	Agrega soporte para PostgreSQL como motor de base de datos de destino. Agrega la posibilidad de generar esquemas convertidos como scripts y de guardar los scripts en archivos antes de aplicar el esquema a la instancia de base de datos de destino.	14 de enero de 2016
1.0.103	Versión de mantenimiento	Agrega capacidad de proyectos sin conexión, la posibilidad de comprobar si hay nuevas versiones y administración de la memoria y el desempeño.	2 de diciembre de 2015
1.0.101	Versión de mantenimiento	Agrega el asistente para crear un nuevo proyecto de migración de base de datos. Agrega la capacidad de guardar el informe de evaluación de la migración de la base de datos en formato PDF.	19 de octubre de 2015
1.0.100	Versión de prueba	Ofrece la guía del usuario para la versión preliminar de la AWS Schema Conversion Tool.	7 de octubre de 2015

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la version original de inglés, prevalecerá la version en inglés.